Bottleneck Bichromatic Full Steiner Trees

A. Karim Abu-Affash* Sujoy Bhore† Paz Carmi‡ Dibyayan Chakraborty§

Abstract

Given two sets of points in the plane, Q of n (terminal) points and S of m (Steiner) points, where each of Q and S contains bichromatic points (red and blue points), a full bichromatic Steiner tree is a Steiner tree in which all points of Q are leaves and each edge of the tree is bichromatic (i.e., connects a red and a blue point). In the bottleneck bichromatic full Steiner tree (BBFST) problem, the goal is to compute a bichromatic full Steiner tree T, such that the length of the longest edge in T is minimized. In k-BBFST problem, the goal is to find a bichromatic full Steiner tree T with at most $k \leq m$ Steiner points from S, such that the length of the longest edge in T is minimized. In this paper, we present an $O((n+m)\log m)$ time algorithm that solves the BBFST problem. Moreover, we show that k-BBFST problem is NP-hard and we give a polynomial-time 9-approximation algorithm for the problem.

1 Introduction

Given a weighted graph $G = (V, E)$ with $V = Q \cup S$, where Q and S are sets of terminal and Steiner points, respectively, a Steiner tree is an acyclic connected subgraph of G spanning all vertices of Q. Informally, Steiner points are new auxiliary nodes that can be added to the network to improve its performance. In the classical *Steiner tree* problem, the goal is to find a Steiner tree T, such that the length of the edges of T is minimized. This problem has been shown to be NP-complete [6, 16], and for arbitrary weighted graphs, many approximation algorithms have been proposed [8, 18, 19].

In the geometric context, i.e., Q and S are disjoint sets of points in the plane, G is the complete graph over $V = Q \cup S$, and the weight of each edge (p, q) in G is the Euclidean distance between p and q. Arora [4] showed that the geometric Steiner tree problem can be efficiently approximated close to optimal.

A Steiner tree is *full* if all terminals are leaves of the tree. In the bottleneck full Steiner tree problem (BFST), the goal is to compute a full Steiner tree with minimum bottleneck (i.e., the length of the longest edge). The k-BFST problem is a restricted version of the BFST problem, for which, in addition to the sets Q and S, we are given a positive integer k, and the goal is to compute a full Steiner tree T with at most k Steiner points such that the bottleneck of T is minimized. Abu-Affash [1] gave a $O((n+m)\log^2 m)$ algorithm for the BFST problem and showed that the k-BFST problem is NP-hard but admits a polynomial-time 4-approximation algorithm. Later, Biniaz et al [10] gave an $O((n+m)\log m)$ algorithm for the BFST problem.

We consider the BFST and the k-BFST problems in bichromatic point sets. Given two sets of points in the plane; a set Q of n red and blue terminals and a set S of m red and blue Steiner points, the goal in the bottleneck bichromatic full Steiner tree (BBFST) problem is to find a full Steiner tree T such that each edge in T connects a red and a blue point and the bottleneck of T is minimized. We refer to this tree as a bichromatic full Steiner tree. In the k-BBFST problem, the goal is to compute a bichromatic full Steiner tree T with at most k Steiner points, such that its bottleneck is minimized, where $k \leq m$ is a given positive integer. The bichromatic input appeared in many geometric problems; for example, red-blue intersection [3], red-blue separation [5, 12, 14, 15], and red-blue connection problems [2, 7, 11].

In this paper, we show how to generalize the algorithms in [1] to solve the BBFST problem and to approximate the k-BBFST problem. More precisely, we present an $O((n+m)\log m)$ algorithm that solves the BBFST problem, we show that the k-BBFST problem is NP-hard, and we give a polynomial-time that approximates it within a factor 9.

2 Exact Algorithm for BBFST

Given a set Q of n red and blue terminals and a set S of m red and blue Steiner points in the plane, we present an $O((n+m)\log m)$ time algorithm that computes a bichromatic full Steiner tree of minimum bottleneck. We refer to such a tree as an optimal bichromatic

*Software Engineering Department, Shamoon College of Engineering, Beer-Sheva 84100, Israel, abuaa1@sce.ac.il.

†Department of Computer Science, Ben-Gurion University, Beer-Sheva 84105, Israel, sujoy.bhore@gmail.com. The research is partially supported by the Lynn and William Frankel Center for Computer Science.

‡Department of Computer Science, Ben-Gurion University, Beer-Sheva 84105, Israel, carmip@cs.bgu.ac.il. The research is partially supported by the Lynn and William Frankel Center for Computer Science.

§Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India, dibyayancg@gmail.com.

full Steiner tree of Q.

Let Q_R and Q_B be the sets of red and blue terminal points of Q, respectively. Similarly, let S_R and S_B be the sets of red and blue Steiner points of S, respectively. We assume that neither S_R nor S_B is empty. Let $MST(S)$ be a minimum-weight bichromatic spanning tree of S (i.e., of the complete bichromatic graph of S_R and S_B). Let $S(T)$ be the set of Steiner points in a bichromatic full Steiner tree T.

Lemma 1 *There exists an optimal bichromatic full Steiner tree T^* of Q, such that $MST(S(T^*))$ is a subtree of $MST(S)$.*

Proof. Let T be an optimal bichromatic full Steiner tree of Q. Let $e = (p_r, p_b)$ be an edge in $MST(S(T))$ but not in $MST(S)$. Let P be the path between p_r and p_b in $MST(S)$. We know that, each edge in P is of length at most $|p_r p_b|$. Moreover, if $T \cup P$ creates a cycle, then this cycle contains e. We add the edges of P to T and we break the produced cycles (by removing the longest edge from each cycle) to obtain a new optimal bichromatic full Steiner tree. By repeating this process for each edge $e \in MST(S(T)) \setminus MST(S)$, we obtain an optimal bichromatic full Steiner tree T^* satisfying the lemma. □

Let $e_1, e_2, \ldots, e_{m-1}$ be the edges of $MST(S)$ sorted in non-decreasing order by their length. For an edge $e_i \in MST(S)$, let $\mathcal{T}_i$ be the forest obtained from $MST(S)$ by deleting all edges of length greater than $|e_i|$ from $MST(S)$. By Lemma 1, there exists an optimal bichromatic full Steiner tree T^* of Q such that $MST(S(T^*))$ is a tree of $\mathcal{T}_i$, for some edge $e_i \in MST(S)$. Thus, by performing binary search on the lengths of edges of $MST(S)$, we can find a forest $\mathcal{T}_i$ that contains a tree T, such that, by connecting each point in Q to its closest point of opposite color in T, we obtain an optimal bichromatic full Steiner tree of Q.

Let λ be the bottleneck of the optimal bichromatic full Steiner tree. For an edge $e_i \in MST(S)$, we decide in $O(n + m)$ time whether $|e_i| > \lambda$ or $|e_i| \leq \lambda$, using the procedure of [10]. (In order to handle the case that $\lambda < |e_1|$ or $\lambda > |e_{m-1}|$, we add the values $|e_0| = 0$ and $|e_m| = \infty$ to the search space.) Therefore, we can find an $0 \leq i \leq m-1$, such that $|e_i| < \lambda \leq |e_{i+1}|$ in $O((n+m)\log m)$ time. If $|e_i| < \lambda < |e_{i+1}|$, then the optimal bichromatic full Steiner tree of Q is obtained by a tree T from the forest $\mathcal{T}_i$; see Figure 1(a). If $\lambda = |e_{i+1}|$, then the optimal bichromatic full Steiner tree of Q is obtained by a tree T from the forest $\mathcal{T}_{i+1}$; see Figure 1(b). Thus, in both cases, we can find the tree T in the set $\mathcal{T}_i \cup \mathcal{T}_{i+1}$, such that, by connecting each terminal in Q to its closest point of opposite color in T, we obtain an optimal bichromatic full Steiner tree of Q. We conclude by the following theorem.

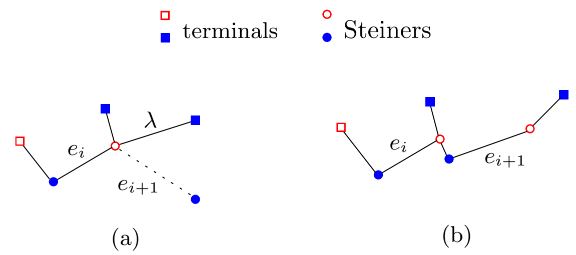

Figure 1: The optimal full bichromatic Steiner tree is obtained (a) from $\mathcal{T}_i$, when $|e_i| < \lambda < |e_{i+1}|$ and (b) from $\mathcal{T}_{i+1}$, when $\lambda = |e_{i+1}|$.

Theorem 2 *The BBFST problem can be solved in $O((n+m)\log m)$ time.*

3 Approximation Algorithm for k-BBFST

Given two sets of points in the plane; a set Q of n red and blue terminal points, a set S of m red and blue Steiner points, and a positive integer $k \leq m$, the goal in the k-BBFST problem is to compute a bichromatic full Steiner tree with at most k Steiner points from S and its bottleneck is minimized. In this section, we first prove that the k-BBFST problem is NP-hard. Then, we present a polynomial-time approximation algorithm with performance ratio 9.

3.1 Hardness proof

We prove the following theorem.

Theorem 3 *The k-BBFST problem is NP-hard.*

Proof. We adopt that proof of Abu-Affash [1] for the k-BFST problem. The proof is based on a reduction from the problem **Connected vertex cover in planar graphs with maximum degree 4** which is NP-complete [17]. Given a planar graph $G = (V, E)$ with vertex degree at most 4 and an integer k, does there exist a vertex cover V^* for G such that $|V^*| \leq k$ and the subgraph of G induced by V^* is connected?

Given a planar graph $G = (V, E)$ with vertex degree at most 4 and an integer k, we construct, in polynomial time, two sets Q and S and compute an integer k', such that G has a connected vertex cover of size at most k if and only if there exists a bichromatic full Steiner tree T of Q with at most k' Steiner points and bottleneck at most 1.

Let $G = (V, E)$ be a planar graph with vertex degree at most 4 and let k be an integer. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$ be the vertices and the edges of G, respectively. We first embed G into a rectangular grid, with distance at least 4 between adjacent vertices. Each vertex $v_i \in V$ corresponds to some grid vertex and each edge $e = (v_i, v_j) \in E$ corresponds to a rectilinear path p_e, consisting of some horizontal and vertical elementary grid segments, whose endpoints are the grid vertices corresponding to v_i and v_j. In addition, these paths are pairwise disjoints; see Figure 2. This embedding can be done in $O(n)$ time and the size of the grid is at most $n-2$ by $n-2$; see [20].

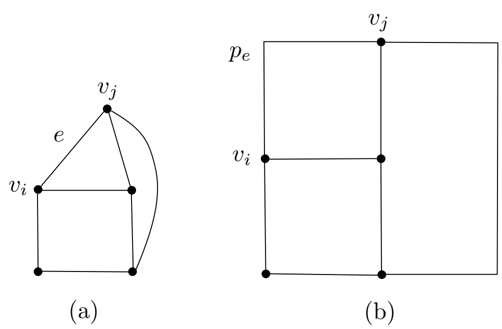

Figure 2: (a) A planar graph $G = (V, E)$, and (b) the embedded graph $G' = (V', E')$ of G.

For each vertex $v_i \in V$ we replace v by a blue Steiner point v'_i; see Figure 3. Let $V' = \{v'_1, v'_2, \ldots, v'_n\}$ be the set of these Steiner points, and let $E' = \{p_{e_1}, p_{e_2}, \ldots, p_{e_m}\}$ be the set of edges (paths) corresponding to the edges of E. We now place two types of points on the interior of each edge $p_e \in E'$. Let $|p_e|$ denote the total length of the grid segments of p_e. We place $|p_e| - 1$ bichromatic Steiner points (red and blue points alternatively) on p_e, such that the distance between any adjacent points is exactly 1, and denote by $s(e)$ this set of Steiner points. Moreover, for each set $s(e)$, we place a red terminal between (in the middle of) every two adjacent points in $s(e)$. Denote by $t(e)$ this set of terminals and notice that $|t(e)| = |p_e| - 2$; see Figure 3. Finally, we set

$$Q = \bigcup_{e \in E} t(e)\,,$$
$$S = V' \cup \bigcup_{e \in E} s(e) \text{ and}$$
$$k' = \sum_{e \in E} |s(e)| - m + 2k - 1\,.$$

For each edge $p_e \in E'$, let $c(e)$ be the set of Steiner points in $s(e)$ except the endpoints, i.e., except the first and the last points. Observe that, connecting every adjacent two Steiner points in $c(e)$ (to form a bichromatic path) and connecting each terminal in $t(e)$ to its closest blue Steiner point in $c(e)$ produces a bichromatic full Steiner tree of $t(e)$ with $|s(e)| - 2$ Steiner points and bottleneck 1. On the other hand, observe that at least $|s(e)| - 2$ Steiner points are necessary to construct a bichromatic full Steiner tree of $t(e)$ with bottleneck at most 1. Denote by T_e such a bichromatic full Steiner tree; see Figure 3.

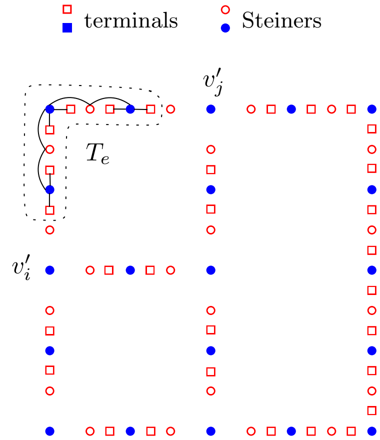

Figure 3: The produced sets: V', $s(e)$, and $t(e)$. T_e is the bichromatic full steiner tree of $t(e)$.

Clearly, the number of points in $Q \cup S$ is $O(n^4)$. Therefore, the reduction can be done in polynomial time. We now prove the correctness of the reduction. Suppose that G has a connected vertex cover V^* with $|V^*| \le k$. We construct a bichromatic full Steiner tree of Q as follows. For each edge $e \in E$, we construct the tree T_e (as described above). Let T' be any spanning tree of the subgraph of G induced by V^*. This spanning tree exists by the connectivity of V^* and contains $|V^*| - 1$ edges. For each edge $e = (v_i, v_j) \in T'$, we connect the corresponding points $v'_i, v'_j \in S$ (by two edges of length 1) to the tree T_e using their adjacent (first and last) points in $s(e)$. And, for each edge $e = (v_i, v_j) \in E \setminus T'$, we select one endpoint v_i of e that belongs to V^* and we connect v'_i (by an edge of length 1) to the tree T_e using its adjacent red Steiner point in $s(e)$. It is easy to see that the constructed tree is a bichromatic full Steiner tree of Q and it has $|V^*| + \sum_{e \in E}(|s(e)| - 2) + 2(|V^*| - 1) + m - (|V^*| - 1) \le \sum_{e \in E} |s(e)| - m + 2k - 1 = k'$ Steiner points and bottleneck exactly 1.

Conversely, suppose that there exists a bichromatic full Steiner tree T of Q with at most k' Steiner points and bottleneck at most 1. Let V^* be the subset of points of V' that appear in T, and let T' be the subtree of T spanning V^*. For each subset $t(e) \subseteq Q$, let T_e be the subtree of T spanning the points in $t(e)$. Since the bottleneck of T is at most 1, (i) by the above observation, T_e contains at least $|s(e)| - 2$ Steiner points, and (ii) each tree T_e is connected to at least one point from V^*, which implies that the set of vertices in G corresponding to the points in V^* is a connected vertex cover of G. Moreover, a tree T_e which is also a subtree of T' is connected to two points from V^* via the endpoints of $s(e)$ (there are $|V^*|-1$ such trees), and a tree T_e which is not a subtree of T' is connected to one point from V^* via one endpoint of $s(e)$ (there are $m-(|V^*|-1)$ such trees). Thus, T contains at least $|V^*|+\sum_{e\in E}(|s(e)|-2)+2(|V^*|-1)+m-(|V^*|-1)$ Steiner points. On the other hand, T contains at most $k'=\sum_{e\in E}|s(e)|-m+2k-1$ Steiner points. This implies that V^* is of size at most k, which completes the proof. ☐

3.2 Approximation algorithm

We devise a polynomial-time approximation algorithm for computing a bichromatic full Steiner tree with at most k Steiner points (k-BFST for short), such that its bottleneck is at most 9 times the bottleneck of an optimal k-BFST.

Let Q_R and Q_B be the sets of red and blue terminal points of Q, respectively. Similarly, let S_R and S_B be the sets of red and blue Steiner points of S, respectively. We assume that S_R and S_B contains at least one red and one blue point, respectively. Let $G=(V,E)$ be the graph with $V=Q\cup S$ and $E=(Q_R\times S_B)\cup(Q_B\times S_R)\cup(S_R\times S_B)$. We assume, w.l.o.g., that $E=\{e_1,e_2,\cdots,e_l\}$, such that $|e_1|\le|e_2|\le\cdots\le|e_l|$. Notice that, the bottleneck of an optimal k-BFST is a length of an edge from E. For an edge e_i, Let $G_i=(V,E_i)$ be the graph, such that $E_i=\{e_j\in E:|e_j|\le|e_i|\}$. We devise a procedure which either constructs a k-BFST of Q in G with bottleneck at most 9 times $|e_i|$ or it says that G_i does not contain a k-BFST of Q.

Let G_i^2 be the 2nd power graph of G_i, i.e., G_i^2 has the same set of vertices as G_i and an edge between two vertices if and only if there is a path that contains at most 2 edges between them in G_i. Let $G_i^2(Q)$ be the sub-graph of G_i^2 induced by Q and let Q' be a maximal independent set in $G_i^2(Q)$. Notice that, since all the edges in E are bichromatic, a red terminal and a blue terminal cannot be connected to a same Steiner point in G_i. Hence, a red terminal and a blue terminal cannot be connected to each other in G_i^2. Thus, if $|Q'|=1$, then Q contains points of one color and we can construct a k-BFST of bottleneck at most $3|e_i|$ as follows. Let p be the only point in Q' and assume, w.l.o.g., that p is a red point. We select a blue Steiner point s that is connected to p in G_i and we connect it to all points of Q. Since there is an edge in G_i^2 between p and each other point $q\in Q$, we have $|pq|\le 2|e_i|$, and therefore, $|sq|\le 3|e_i|$.

Thus, we assume that $|Q'|>1$. For any two points $p,q\in Q$, let $\delta_i(p,q)$ be the path between p and q in G_i that contains minimum number of Steiner points. Let $G'=(Q',E')$ be the complete graph over Q'. For each edge (p,q) in E', we assign a weight $w(p,q)$ which is equal to the number of Steiner points in $\delta_i(p,q)$. Let $MST(G')$ be the minimum spanning tree of G' under w. We define the normalized weight of $MST(G')$ as $W(MST(G'))=\sum_{e\in MST(G')}\lfloor w(e)/2\rfloor$.

Lemma 4 *If G_i contains a k-BFST of Q', then $W(MST(G'))\le k$*

Proof. Let T be a k-BFST of Q' in G_i. We construct a spanning tree T' of G' such that $W(T')\le k$. We start by T and we transform it into T' by an iterative process. We start by selecting an arbitrary Steiner point as the root of T; see Figure 4. In each iteration, we select the deepest leaf p in the rooted tree, which is a terminal, and we connect it to its closest terminal q by an edge (p,q) of weight equal to the number of Steiner points between them. Let s be the lowest common ancestor of p and q. We then remove the Steiner points between p and s. In the last iteration, we remove all of the remaining points.

For example, in Figure 4, we show a construction of T' from T. In iteration 1, we select p_1, connect it to p_2 by an edge of weight 4 and remove the points between p_1 and s_1. In iteration 2, we select p_3, connect it to p_4 by an edge of weight 4, and remove the points between p_3 and s_2. In iteration 3, we select p_6, connect it to p_5 by an edge of weight 3, and remove the points between p_6 and s_3. In iteration 4, we select p_5, connect it to p_4 by an edge of weight 6, and remove the points between p_5 and s_4. In the last iteration, we select p_2, connect it to p_4 by an edge of weight 5, and remove the all the remaining points between p_2 and p_4.

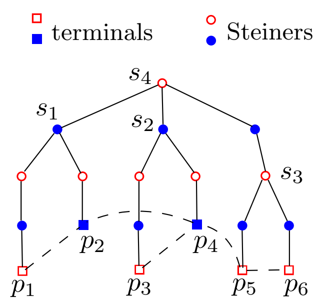

Figure 4: Constructing T' from T.

Since, in each iteration, we select the deepest terminal, we add to T' an edge (p,q) of weight $w(p,q)$, and we remove at least $\lfloor w(p,q)/2\rfloor$ Steiner points from T. Thus, we have $W(T')=\sum_{e\in T'}\lfloor w(e)/2\rfloor\le k$. Finally, since T' is also a spanning tree of G', we have $W(MST(G'))\le W(T')\le k$. ☐

We now describe the algorithm. For each edge $e_i\in E$ in the sorted order, we construct the graphs G_i, G_i^2, and $G_i^2(Q)$. Then, we compute a maximal independent set Q' in $G_i^2(Q)$. If $|Q'|=1$, then we construct a k-BFST of Q with bottleneck at most 3 times $|e_i|$. Otherwise, we construct the complete graph G' over Q', and we compute a minimum spanning tree $MST(G')$ of G' with respect to the weight function w. If $W(MST(G')) > k$, then we proceed to the next edge e_{i+1}. Otherwise, we construct a k-BFST of Q with bottleneck at most 9 times $|e_i|$ as follows.

For each edge $(p, q) \in T$, there is a bichromatic path $\delta_i(p, q)$ between p and q in G_i that contains $w(p, q)$ Steiner points. We select $\lfloor w(p, q)/2 \rfloor$ Steiner points on any shortest Steiner path between p and q in G_i by the following procedure.

We select an arbitrary leaf p in $MST(G')$ and we traverse $MST(G')$ starting from p. Let q be the point that is connected to p in $MST(G')$. Set $S' = \emptyset$. We call the recursive procedure $SelectSteiners(p, q, color(p), S')$ (Procedure 1) that selects at most k Steiner points and adds them to S'; see also Figure 5.

Procedure 1 $SelectSteiners(p, q, color, S')$

$j \leftarrow w(p, q)$
let $s_1, s_2, \ldots, s_j$ be the Steiner points in $\delta_i(p, q)$
$x \leftarrow 0$
if $color(s_1) \neq color$ **then**
    $i \leftarrow 1$
else
    $i \leftarrow 2$
while $i + 3x \leq j$ **do**
    $S' \leftarrow S' \cup \{s_{i+3x}\}$
    $x \leftarrow x + 1$
for each $(q, t) \in MST(G')$, such that $t \neq p$ **do**
    $SelectSteiners(q, t, color(s_{i+3(x-1)}), S')$

It is not hard to see that for each edge (p, q) in $MST(G')$, we add to S' at most $\lfloor w(p, q)/2 \rfloor$ Steiner points. Therefore, $|S'| \leq k$. Next, we construct a minimum-weight bichromatic spanning tree $MST(S')$ of S' (i.e., of the complete bichromatic (Euclidean) graph over S'). Notice that, each edge in $MST(S')$ is of length at most $5|e_i|$; see Figure 5. Finally, we connect each terminal in Q to its nearest opposite color Steiner point in S' to obtain a bichromatic full Steiner tree. This guarantees that each terminal in Q' is connected to a Steiner point with an edge of length at most $7|e_i|$; see Figure 5, and each terminal in $Q \setminus Q'$ is connected to a Steiner point with an edge of length at most $9|e_i|$.

Remark. If Q' contains only one red and one blue points p and q, respectively, $k = 2$, and $MST(G')$ is a path between p and q that contains exactly 2 Steiner points, a blue Steiner point s_1 and a red Steiner point s_2, then we construct a k-BFST by connecting all the points in Q_R to s_1 and all the points in Q_B to s_2. This k-BFST contains exactly 2 Steiner points and its bottleneck is at most $3|e_i|$.

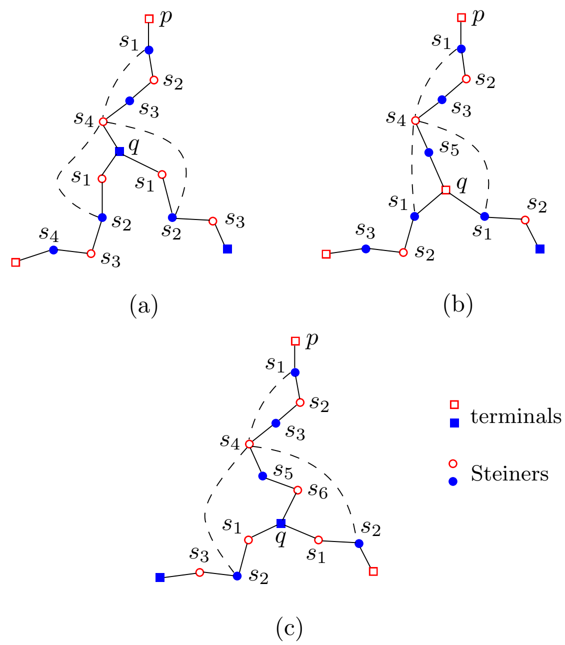

Figure 5: Illustrating the selection of the Steiner points in Procedure 1.

Lemma 5 *Our algorithm constructs a k-BFST of Q with bottleneck at most 9 times the bottleneck of an optimal k-BFST.*

Proof. Let $e_i \in E$ be the first edge satisfying $W(T) \leq k$. Thus, by Lemma 4, the bottleneck of any k-BFST in G is at least $|e_i|$. Therefore, the constructed k-BFST has a bottleneck at most 9 times the bottleneck of an optimal k-BFST. □

Lemma 6 *Our algorithm runs in polynomial time.*

Proof. Notice that, for each edge $e_i \in E$, the third power graph G_i^2 is of size $O((n+m)^2)$. Thus, G_i^2 can be computed from G_i in $O((n+m)^2)$ time, and computing a maximal independent set Q' in $G_i^2(Q)$ also takes $O((n+m)^2)$ time. The construction of G' on Q' can be done in $O((n+m)^3)$ time, by computing the shortest Steiner paths between each pair of points in Q' [13]. Computing a minimum spanning tree of G' can be done in $O(n^2)$ time. Procedure 1 runs in $O(k(n+m))$ time. the construction of the obtained full Steiner tree can be done in $O((n+k)\log k)$. Therefore, the algorithm runs in polynomial time. □

The following theorem summarizes the result of this section.

Theorem 7 *The above algorithm computes a k-BFST with bottleneck at most 9 times the bottleneck of an optimal k-BFST in polynomial time.*

References

[1] A. K. Abu-Affash. The Euclidean bottleneck full steiner tree problem. *Algorithmica*, 71(1):139–151, 2015.

[2] P. K. Agarwal, H. Edelsbrunner, and O. Schwarzkopf. Euclidean minimum spanning trees and bichromatic closest pairs. *Disc. & Comput. Geom.*, 6:407–422, 1991.

[3] P. K. Agarwal and M. Sharir. Red-blue intersection detection algorithms, with applications to motion planning and collision detection. *SIAM J. Comput.*, 19(2):297–321, 1990.

[4] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[5] S. Arora and K. L. Chang. Approximation schemes for degree-restricted MST and red-blue separation problem. In *Proceedings of ICALP*, pages 176–188, 2003.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[7] M. J. Atallah and D. Z. Chen. On connecting red and blue rectilinear polygonal obstacles with nonintersecting monotone rectilinear paths. *Int. J. Comput. Geom. Appl.*, 11(4):373–400, 2001.

[8] P. Berman and V. Ramaiyer. Improved approximations for the steiner tree problem. In *Proceedings of SODA*, pages 325–334, 1992.

[9] A. Biniaz, P. Bose, D. Eppstein, A. Maheshwari, P. Morin, and M. Smid. Spanning Trees in Multipartite Geometric Graphs. *CoRR*, abs/1611.01661, 2016.

[10] A. Biniaz, A. Maheshwari, and M. Smid. An optimal algorithm for the Euclidean bottleneck full steiner tree problem. *Comput. Geom.*, 47(3):377–380, 2014.

[11] A. Biniaz, A. Maheshwari, and M. Smid. Bottleneck bichromatic plane matching of points. In *Proceedings of CCCG*, 2014.

[12] J.-D. Boissonnat, J. Czyzowicz, O. Devillers, J. Urrutia, and M. Yvinec. Computing largest circles separating two sets of segments. *Int. J. Comput. Geom. Appl.*, 10(1):41–53, 2000.

[13] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms, 3rd edition*. The MIT Press, 2009.

[14] E. D. Demaine, J. Erickson, F. Hurtado, J. Iacono, S. Langerman, H. Meijer, M. H. Overmars, and S. Whitesides. Separating point sets in polygonal environments. *Int. J. Comput. Geom. Appl.*, 15(4):403–420, 2005.

[15] H. Everett, J.-M. Robert, and M. J. van Kreveld. An optimal algorithm for the (<= k)-levels, with applications to separation and transversal problems. *Int. J. Comput. Geom. Appl.*, 6(3):247–261, 1996.

[16] M. R. Garey, R. L. Graham, and D. S. Johnson. The complexity of computing steiner minimal trees. *SIAM J. Appl. Math.*, 32(4):835–859, 1977.

[17] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM J. Appl. Math.*, 32(4):826–834, 1977.

[18] M. Karpinski and A. Zelikovsky. New approximation algorithms for the steiner tree problems. *J. Comb. Opt.*, 1(1):47–65, 1997.

[19] H. J. Prömel and A. Steger. A new approximation algorithm for the steiner tree problem with performance ratio 5/3. *Journal of Algorithms*, 36(1):89–101, 2000.

[20] W. Schnyder. Embedding planar graphs on the grid. In *Proceedings of SODA*, pages 138–148, 1990.