Approximating longest spanning tree with neighborhoods

Ahmad Biniaz

School of Computer Science, University of Windsor
ahmad.biniaz@gmail.com

Abstract

We study the following maximization problem in the Euclidean plane: Given a collection of neighborhoods (polygonal regions) in the plane, the goal is to select a point in each neighborhood so that the longest spanning tree on selected points has maximum length. It is not known whether or not this problem is NP-hard. We present an approximation algorithm with ratio 0.548 for this problem. This improves the previous best known ratio of 0.511 due to Chen and Dumitrescu (Discrete Mathematics, Algorithms and Applications, 2020).

The presented algorithm takes linear time after computing a diameter of the neighborhoods. Even though our algorithm itself is fairly simple, its analysis is rather involved. In some part we deal with a minimization problem involving multiple parameters. We use a sequence of geometric transformations to reduce the number of parameters and simplify the analysis.

1 Introduction

The spanning tree is a well-studied and fundamental structure in graph theory and combinatorics. The well-known minimum spanning tree (Min-ST) problem asks for a spanning tree with minimum total edge-weight. In contrast, the maximum spanning tree (Max-ST) problem asks for a spanning tree with maximum total edge-weight. In the context of abstract graphs, the two problems are algorithmically equivalent in the sense that an algorithm that finds a Min-ST can also find a Max-ST within the same time bound (by simply negating the edge weights), and vice versa. The situation is quite different in the context of geometric graphs where vertices are points in the plane and edge-weights are Euclidean distances between points. In geometric graphs, an algorithm that exploits geometry for finding a Min-ST is not necessarily useful for finding a Max-ST because there is no known geometric transformation between the "nearest" and "farthest" relations among points [22]. The existing geometric algorithms, for solving these two problems, exploit different sets of techniques.

Problems related to maximum configurations in the plane (also know as *long configurations*) have received considerable attention after the seminal work of Alon, Rajagopalan, and Suri [2]. They studied configurations such as spanning trees, perfect matchings and Hamiltonian paths. In this paper we study the *longest spanning tree with neighborhoods* (Max-ST-NB) problem. We are given a collection of n neighborhoods (polygonal regions) in the Euclidean plane and we want to find a maximum-length tree that connects n representative points, one point from each neighborhood, as in Figure 1(a). The *length* of the tree is the total Euclidean length of its edges. Each *neighborhood* is the union of simple polygons, and the neighborhoods are not necessarily disjoint. The neighborhoods

This research is supported by NSERC.

are assumed to be *colored* by n different colors. The classical Euclidean Max-ST problem is a special case of the Max-ST-NB in which each neighborhood consists of exactly one point, as in Figure 1(b). Although the Euclidean Max-ST problem can be solved in $O(n \log n)$ time [22], it is not known whether or not the Max-ST-NB problem can be solved in polynomial time. The difficulty lies in choosing representative points from neighborhoods; once these points are selected, the problem is reduced to the Euclidean Max-ST problem.

(a) Max-ST-NB

(b) Max-ST

Figure 1: (a) Longest spanning tree with four polygonal neighborhoods that are colored red, green, blue, and purple. (b) Euclidean maximum spanning tree.

It is easily seen (see Section 2) that the longest star (a *star* is a tree that connects a point in one neighborhood to a point in each of the other neighborhoods) achieves a 0.5-approximate solution for the Max-ST-NB problem. Recently, Chen and Dumitrescu [10] present an approximation algorithm with ratio 0.511, which is the first improvement beyond 0.5.

Although any optimal solution for the Max-ST problem contains a diametral pair (two points with maximum distance) as an edge, an optimal solution for the Max-ST-NB problem does not necessarily contain any bichromatic diametral pair (two points with maximum distance that belong to different neighborhoods). Another result of Chen and Dumitrescu [10] shows that any algorithm, that always includes a bichromatic diametral pair in the solution, cannot achieve an approximation ratio better than $\sqrt{2-\sqrt{3}} \approx 0.517$. This somehow breaks the hope for getting a good approximation ratio by using greedy techniques. Thus, to improve the ratio beyond 0.517 one needs to employ some nontrivial ideas.

1.1 Our contribution and approach

We present an approximation algorithm for the Max-ST-NB problem with improved ratio $\frac{\sqrt{7}-1}{3} \approx 0.548$. Our algorithm is not complicated: We compute a double-star (a tree of diameter 3 in which every vertex is connected to one of the two vertices that are considered as centers) centered at a bichromatic diametral pair, and compute up to three stars (trees of diameter 2) centered at points on the smallest enclosing circle, and then report the longest one; see Figure 2. Our algorithm takes linear time after computing a bichromatic diameter. Our analysis involves a minimization problem with multiple variables. To simplify the analysis we apply a sequence of geometric transformations and reduce the number of variables. The following theorem summarizes our main contribution.

Theorem 1. *A 0.548-approximation for the longest spanning tree with neighborhoods can be computed in linear time after computing a bichromatic diameter.*

As a minor result we improve the upper bound 0.517 on the approximation ratio of algorithms that always include a bichromatic diameter in their solutions. We show that the ratio of such algorithms cannot be better than 0.5. This upper bound is tight because there exists a 0.5-approximation algorithm that always includes a bichromatic diameter (see Section 2). Therefore, to obtain a ratio of better than 0.5 one should take into account also spanning trees that do not contain any bichromatic diameter. Indeed the output of our 0.548-approximation algorithm does not necessarily contain a bichromatic diameter.

1.2 Related problems and applications

The Max-ST-NB problem has the same flavor as the Euclidean group Steiner tree problem in which we are given n groups of points in the plane and the goal is to find a shortest tree that contains "at least" one point from each group. The group Steiner tree problem in graphs is NP-hard and cannot be approximated by a factor $O(\log^{2-\epsilon} n)$ for any $\epsilon > 0$ [16]. The Max-ST-NB also lies in the concept of *imprecision* in computational geometry where each input point is provided as a region of uncertainty and the exact position of the point may be anywhere in the region; see for example [13, 18]. Analogous problems have been studied for other structures, e.g., minimum spanning tree with neighborhoods [8, 13, 25], traveling salesman tour with neighborhoods [3, 20, 21] (which is APX-hard [12]), and convex hull of imprecise points [18, 23], to name a few. We refer the interested readers to the thesis of Löffler [17].

The maximum spanning tree and related problems, in addition to their fundamental nature, find applications in worst-case analysis of various heuristics in combinatorial optimization [2], and in appròminating maximum triangulations [5, pp. 338]. They also appear in clustering algorithms where one needs to partition a set of entities into well-separated and homogeneous clusters [4, 22]. Maximum spanning trees are directly related to computing diameter and farthest neighbors which are fundamental problems in computational geometry, with many applications [1].

2 Preliminaries for the algorithm

The output of our algorithm is either a star or a double-star. A *star*, centered at a vertex p, is a tree in which every edge is incident to p. A *double-star*, centered at two vertices p and q, is a tree that contains the edge pq and its every other edge is incident to either p or q.

Let P be a set of points in the Euclidean plane. The *smallest enclosing disk* for P is the smallest disk that contains all the points of P. A *diametral pair* of P is a pair of points in P that attain the maximum Euclidean distance. If the points in P are colored, then a *bichromatic diametral pair* of P is defined as a pair of points in P with different colors that attain the maximum Euclidean distance. The *center of mass* of P (also knows as the *centroid*) is a point m in the plane such that for any arbitrary point u in the plane we have

$$\sum_{p\in P} \overrightarrow{up} = |P| \cdot \overrightarrow{um}, \tag{1}$$

where $\overrightarrow{up}$ is the position vector of p relative to u. Intuitively, the center of mass of a system of weighted points is a point where the system balances (in our case, we assume that all points of P have the same weight).

The intersection of two disks is called a *lens*. We denote the straight line segment between two points p and q in the plane by pq and we denote the Euclidean distance between them by $|pq|$. In our context, a *geometric graph* is a graph whose vertices are points in the plane and whose edges are straight line segments. The length of a geometric graph G, denoted by $\text{len}(G)$, is the total Euclidean length of its edges.

A simple 0.5-approximation algorithm. Chen and Dumitrescu [10] pointed out the following simple 0.5-approximation algorithm for the Max-ST-NB problem (a similar approach was previously used in [2] and [14]). Take a bichromatic diametral pair (a, b) from the vertices of the given n polygonal neighborhoods; a and b belong to two different neighborhoods. Choose an arbitrary point from each of the other $n-2$ neighborhoods. Let S_a be the star obtained by connecting a to b and a to all chosen points. Define S_b analogously on the same point set. Every edge of any optimal solution T^* has length at most $|ab|$, and thus $\text{len}(T^*) \leqslant (n-1)|ab|$. By the triangle inequality $\text{len}(S_a) + \text{len}(S_b) \geqslant n|ab| \geqslant \text{len}(T^*)$. Therefore the longer of S_a and S_b is a 0.5-approximate solution for the problem.

3 The approximation algorithm

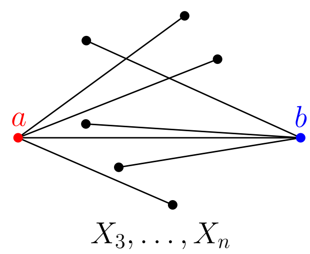
In this section we prove Theorem 1. Put $\delta = \frac{\sqrt{7}-1}{3} \approx 0.548$. To facilitate comparisons we use the same notation as of Chen and Dumitrescu [10]. Let $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$ be the given collection of n polygonal neighborhoods of total N vertices. We assume that each X_i is colored by a unique color. Our algorithm selects representative points only from boundary vertices of the polygonal neighborhoods. Thus, in the algorithm (but not in the analysis) we consider each polygonal neighborhood X_i as the set of its boundary vertices, and consequently we consider $\mathcal{X}$ as a collection of N points colored by n colors. Define the *longest spanning star* centered at a point $p \in X_i$ as the star connecting p to its farthest point in every other neighborhood.

Algorithm. The main idea of the algorithm is simple: We compute a spanning double-star D and (at most) three spanning stars S_1, S_2, S_3, and then return the longest one.

We compute D as follows. Let (a, b) be a bichromatic diametral pair of $\mathcal{X}$. After a suitable relabeling assume that $a \in X_1$ and $b \in X_2$. Add the edge ab to D. For each X_i, with $i \in \{3, \dots, n\}$, find a vertex $p_i \in X_i$ that is farthest from a and find a vertex $q_i \in X_i$ that is farthest from b (it might be the case that $p_i = q_i$). If $|ap_i| \geqslant |bq_i|$ then add ap_i to D otherwise add bq_i to D. The double-star D spans all neighborhoods in $\mathcal{X}$, and each edge of D has length at least $|ab|/2$ because $|ap_i| + |bq_i| \geqslant |ab|$ due to our choices of p_i and q_i; see Figure 2(a). Next we introduce the stars S_1, S_2, and S_3. Let C be the smallest enclosing disk for $\mathcal{X}$. Notice that the boundary of C contains at least two points of $\mathcal{X}$. If it contains exactly two points then we define S_1 and S_2 as the two longest spanning stars that are centered at these points (in this case we do not have S_3). If it contains three or more points then there exist three of them such that the triangle formed by those points contains the center of C [11, Chapter 4, Section 4.7]. In this case we define S_1, S_2, and S_3 as the three longest spanning stars that are centered at those three points, as in Figure 2(b).

Notice that, although the double-star D contains the bichromatic diameter ab, the stars S_1, S_2, and S_3 may not contain any bichromatic diameter.

Running time. The smallest enclosing disk C for $\mathcal{X}$ can be computed in $O(N)$ time [9, 19, 24]. The result of [7], that computes a maximum spanning tree on multicolored points, implies that a bichromatic diametral pair (a, b) for $\mathcal{X}$ can be found in $O(N \log N \log n)$ time (the algorithm of Bhattacharya and Toussaint [6] also computes a bichromatic diameter, but only for two-colored points). The rest of our algorithm (finding farthest points from a, b, and from the points on the boundary of C) takes $O(N)$ time.



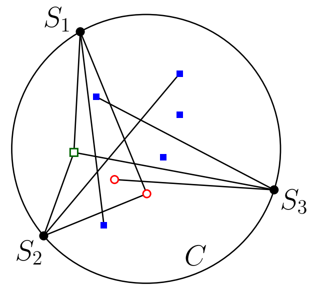


Figure 2: Illustration of the algorithm: (a) the double-star D, and (b) longest stars S_1, S_2, and S_3.

3.1 Analysis of the approximation ratio

Our main plan for analysis works as follows: We show that if the radius of C is at least δ then one of the stars S_i is a desired tree, otherwise the double-star D is a desired tree.

For the analysis we consider $\mathcal{X}$ as the initial collection of polygonal neighborhoods. Let T^* denote a longest spanning tree with neighborhoods in $\mathcal{X}$. It is not hard to see that for any point in the plane, its farthest point in a polygon P must be a vertex of P (see also [11, Chapter 7, Section 7.4]). Thus, any bichromatic diameter of $\mathcal{X}$ is introduced by two vertices of polygons in $\mathcal{X}$. Hence the pair (a, b), selected in the algorithm, is a bichromatic diameter of the initial collection $\mathcal{X}$. Therefore, $|ab|$ is an upper bound for the length of edges in T^*. After a suitable scaling assume that $|ab| = 1$. Since T^* has $n-1$ edges,

$$\text{len}(T^*) \leqslant (n-1)|ab| = n-1. \tag{2}$$

Recall the smallest enclosing disk C from the algorithm. Let c denote the center of C and r denote its radius. If the boundary of C has exactly two points of $\mathcal{X}$ then denote them by c_1 and c_2. In this case the segment c_1c_2 is a diameter of C; see [11, Chapter 4, Section 4.7]. If the boundary of C has three or more points of $\mathcal{X}$ then denote the three points (that are chosen in the algorithm) by c_1, c_2, and c_3. In this case $c_1c_2c_3$ is an acute or a right triangle and the center c lies in its interior or on its boundary; see [11, Chapter 4, Section 4.7]. Recall the longest spanning stars S_1, S_2, S_3 from the algorithm. After a suitable relabeling assume that the star S_i is centered at the point c_i.

Lemma 1. *If $r \geqslant \delta$ and the boundary of C contains exactly two points of $\mathcal{X}$ then*

$$\max\{\text{len}(S_1), \text{len}(S_2)\} \geqslant \delta \cdot \text{len}(T^*).$$

Proof. In this case c_1c_2 is a diameter of C, and thus $|c_1c_2| = 2r \geqslant 2\delta > 1$. As the bichromatic diameter is $|ab| = 1$ and $|c_1c_2| > 1$, it turns out that c_1 and c_2 have the same color. Assume that $c_1, c_2 \in X_1$. Pick an arbitrary point p_i from each X_i with $i \in \{2, \ldots, n\}$. Let S'_1 be the spanning star that connects c_1 to all p_is. Let S'_2 be the spanning star that connects c_2 to all p_is. Notice that $\text{len}(S'_1) \leqslant \text{len}(S_1)$ and $\text{len}(S'_2) \leqslant \text{len}(S_2)$. By bounding the maximum of two numbers by their average, then using the triangle inequality and (2) we get:

$$\begin{aligned}
\max\{\text{len}(S_1), \text{len}(S_2)\} &\geqslant \max\left\{\text{len}(S'_1), \text{len}(S'_2)\right\} \geqslant \frac{1}{2}\left(\text{len}(S'_1) + \text{len}(S'_2)\right) \\
&= \frac{1}{2}\sum_{i=2}^{n}(|c_1p_i| + |p_ic_2|) \geqslant \frac{1}{2}\sum_{i=2}^{n}|c_1c_2| \\
&= \frac{|c_1c_2|}{2} \cdot (n-1) \geqslant \delta \cdot (n-1) \geqslant \delta \cdot \text{len}(T^*). \qquad \square
\end{aligned}$$

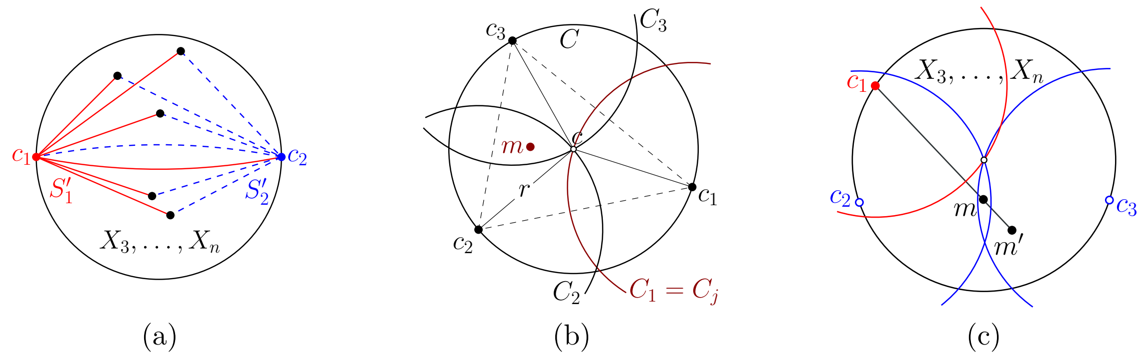


Figure 3: Illustration of the proofs of (a) Lemma 1, (b) Lemma 2, and (c) Lemma 3.

Lemma 2. *If the boundary of C contains three or more points of $\mathcal{X}$ then for any point m in the plane there exists a point $c_j \in \{c_1, c_2, c_3\}$ such that $|c_j m| \geqslant r$.*

Proof. Let C_i be the disk of radius r centered at each c_i. The boundary of each C_i passes through the center c of C, as in Figure 3(b). As the triangle $c_1c_2c_3$ contains c it holds that $C_1 \cap C_2 \cap C_3 = \{c\}$. Therefore there exists a disk C_j that does not have m in its interior, and thus $|c_j m| \geqslant r$. □

Lemma 3. *If $r \geqslant \delta$ and the boundary of C contains three or more points of $\mathcal{X}$ then*

$$\max\{\mathrm{len}(S_1), \mathrm{len}(S_2), \mathrm{len}(S_3)\} \geqslant \delta \cdot \mathrm{len}(T^*).$$

Proof. Recall that the triangle $c_1c_2c_3$ contains the center c of C. We consider three cases depending on similarity of colors of c_1, c_2, and c_3.

- The points c_1, c_2, c_3 have pairwise distinct colors. Thus, they belong to three different neighborhoods. After a suitable relabeling assume that $c_1 \in X_1$, $c_2 \in X_2$, and $c_3 \in X_3$. Pick an arbitrary point from each X_i with $i \in \{4, \dots, n\}$. Denote the selected points by P. Let m be the center of mass of P. By Lemma 2 there exists a point $c_j \in \{c_1, c_2, c_3\}$ where $|c_j m| \geqslant r$. After a suitable relabeling assume that $c_j = c_1$, and thus $|c_1 m| \geqslant r$. By (1) we get

$$\sum_{p \in P} |c_1 p| \geqslant |P| \cdot |c_1 m| \geqslant (n-3) \cdot r \geqslant (n-3) \cdot \delta.$$

  Let S'_1 be the star that connects c_1 to all points of P and to c_2 and c_3. Since S_1 is the longest spanning star centered at c_1, we have that $\mathrm{len}(S_1) \geqslant \mathrm{len}(S'_1)$. Since the triangle $c_1c_2c_3$ contains c, we have $|c_2c_1| + |c_3c_1| \geqslant |c_2c| + |c_3c| = 2r \geqslant 2\delta$. These inequalities and (2) give

$$\mathrm{len}(S_1) \geqslant \mathrm{len}(S'_1) = |c_1c_2| + |c_1c_3| + \sum_{p \in P} |c_1 p| \geqslant 2\delta + (n-3) \cdot \delta = (n-1) \cdot \delta \geqslant \delta \cdot \mathrm{len}(T^*).$$

- All points c_1, c_2, c_3 have the same color. Assume that $c_1, c_2, c_3 \in X_1$. Pick an arbitrary point from each X_i with $i \in \{2, \dots, n\}$. Denote the selected points by P. Let m be the centroid of P, and let c_1 be the point in $\{c_1, c_2, c_3\}$ for which $|c_1 m| \geqslant r$ (by Lemma 2 such a point exists). Let S'_1 be the star that connects c_1 to all points of P. Similar to the previous case by using (1) we get

$$\mathrm{len}(S_1) \geqslant \mathrm{len}(S'_1) \geqslant \sum_{p \in P} |c_1 p| \geqslant |P| \cdot |c_1 m| \geqslant (n-1) \cdot r \geqslant (n-1) \cdot \delta \geqslant \delta \cdot \mathrm{len}(T^*).$$

- Only two of c_1, c_2, c_3 have the same color. This case is depicted in Figure 3(c). Assume that c_2 and c_3 have the same color and they belong to X_2. Also assume that $c_1 \in X_1$. We handle this case in a slightly different way; this is because if the point c_j from Lemma 2 (which would have distance at least r to the centroid) belongs to $\{c_2, c_3\}$ then there is no guarantee that $|c_j c_1| \geqslant \delta$, and hence we may not be able to establish the lower bound $\delta \cdot (n-1)$.

  Pick an arbitrary point from each X_i with $i \in \{3, \dots, n\}$. Let P be the set containing all selected points together with the point c_1. Let m be the centroid of P. Consider the point c_j (from Lemma 2) for which $|c_j m| \geqslant r$. If $c_j = c_2$ then let S'_2 be the star that connects c_2 to all points of P. In this case

$$\mathrm{len}(S_2) \geqslant \mathrm{len}(S'_2) = \sum_{p \in P} |c_2 p| \geqslant |P| \cdot |c_2 m| \geqslant (n-1) \cdot r \geqslant (n-1) \cdot \delta \geqslant \delta \cdot \mathrm{len}(T^*).$$

  If $c_j = c_3$ then by a similar argument we get $\mathrm{len}(S_3) \geqslant \delta \cdot \mathrm{len}(T^*)$.

  Now assume that $c_j = c_1$, and thus $|c_1 m| \geqslant r$. Let $P' = P \setminus \{c_1\}$, and let m' be the centroid of P'. Using the recursive definition of centroid [15][1] (based on the Euclidean rule of the lever) the point m lies on the segment $c_1 m'$, as in Figure 3(c). Informally speaking, if we remove c_1 from P then its (new) centroid moves away from c_1. Therefore $|c_1 m'| \geqslant |c_1 m| \geqslant r$. Let S'_1 be the star obtained by connecting c_1 to all points of P' and to the one of c_2 and c_3 that is farther from c_1. Assume that c_2 is the farther one, and notice that $|c_1 c_2| \geqslant r$. Then,

$$\mathrm{len}(S_1) \geqslant \mathrm{len}(S'_1) = |c_1 c_2| + \sum_{p \in P'} |c_1 p| \geqslant r + |P'| \cdot |c_1 m'| \geqslant r + (n-2) \cdot r \geqslant \delta \cdot \mathrm{len}(T^*). \quad \square$$

Lemmas 1 and 3 take care of our analysis for the case where the radius r of C is at least δ. The next lemma takes care of the case where $r \leqslant \delta$ by showing that in this case the double-star D is a desired tree. We employ a collection of geometric transformations to simplify the proof.

Lemma 4. *If $r \leqslant \delta$ then* $\mathrm{len}(D) \geqslant \delta \cdot \mathrm{len}(T^*)$.

Proof. Recall (a, b) as a bichromatic diametral pair of $\mathcal{X}$. Also recall our assumptions that $a \in X_1$, $b \in X_2$, and that $|ab| = 1$.

One challenge that we face here is that the vertices of our double-star D could be different from the vertices of the optimal tree T^*; this could make it difficult to obtain a lower bound for the length of D in terms of the length of T^*. But we know that the vertices of both D and T^* come from the same ground sets $X_1, \dots, X_n$. Our plan is to compare the length of D with the length of T^* by comparing the lengths of their edges separately. For each $i \in \{1, \dots, n\}$ let p_i^* and p_i be the vertices of T^* and D that belong to X_i, respectively (it might be that $p_i^* = p_i$). Notice that $a = p_1$ and $b = p_2$. Direct all edges of T^* towards p_1^* and direct all edges of D towards p_1. To each vertex of T^* and D (except p_1^* and p_1) assign its unique outgoing edge. For each $i \in \{2, \dots, n\}$ let $\mathrm{len}(p_i^*)$ and $\mathrm{len}(p_i)$ be the length of edges that are assigned to p_i^* and p_i, respectively. We already know that $\mathrm{len}(p_2) = |ab| = 1$ and $\mathrm{len}(p_i^*) \leqslant |ab| = 1$ for all i. Thus, in order to show that $\mathrm{len}(D) \geqslant \delta \cdot \mathrm{len}(T^*)$ it suffices to show that

$$\frac{\mathrm{len}(p_i)}{\mathrm{len}(p_i^*)} \geqslant \delta, \tag{3}$$

for each $i \in \{3, \dots, n\}$. From the optimization point of view, we are interested in the minimum value of the ratio $\frac{\mathrm{len}(p_i)}{\mathrm{len}(p_i^*)}$ over all pairs (p_i, p_i^*). In particular we want this value to be at least δ.

[1] To determine the centroid of $k > 2$ points we can replace any $k-1$ of them by their centroid and then find the centroid of the two remaining points.

From now on we consider a fixed value of $i \in \{3, \dots, n\}$. For simplicity we write p for p_i, p^* for p_i^*, and X for X_i. In the rest of this section we will show that $\text{len}(p) \geqslant \delta \cdot \text{len}(p^*)$. Recall from the algorithm that p is connected to the farther of a and b, and thus $\text{len}(p) = \max\{|pa|, |pb|\}$. Let $D(a, \delta)$ and $D(b, \delta)$ be the disks of radii δ that are centered at a and b, respectively. If p is outside $D(a, \delta)$ then $\text{len}(p) \geqslant |pa| \geqslant \delta \geqslant \delta \cdot \text{len}(p^*)$. Likewise, if p is outside $D(b, \delta)$ then $\text{len}(p) \geqslant \delta \cdot \text{len}(p^*)$, and we are done.

In the rest of this section we assume that p is in the lens $L = D(a, \delta) \cap D(b, \delta)$ which is depicted in Figure 4(a). In the current setting, the neighborhood X (which contains p) lies entirely in L because otherwise our algorithm would have picked a point of X outside L. Therefore, the point p^* (which also belongs to X) lies in L. Moreover $\max\{|ap|, |bp|\} \geqslant \max\{|ap^*|, |bp^*|\}$ because otherwise our algorithm would have picked p^* instead of p. Thus, to achieve (3), it suffices to show that

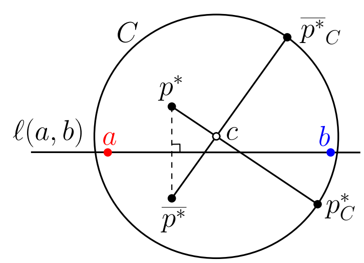
$$\frac{\max\{|ap^*|, |bp^*|\}}{\text{len}(p^*)} \geqslant \delta. \tag{4}$$

For any point q in disk C let q_C be the intersection point of the boundary of C with the ray emanating from q and passing through the center c; Figure 4(a) depicts this for point $q = p^*$. The point q_C is the farthest point of C from q. Thus, the largest possible length of the edge of T^* that is assigned to p^* is $|p^* p^*_C|$, that is, $\text{len}(p^*) \leqslant |p^* p^*_C|$. Thus, to achieve (4), it suffices to show that

$$\frac{\max\{|ap^*|, |bp^*|\}}{|p^* p^*_C|} \geqslant \delta. \tag{5}$$

Inequality (5) deals with a minimization problem which has multiple variables, including the coordinates of a, b, p^*, and c. We use a sequence of geometric transformations to reduce the number of variables and simplify the analysis. Our transformations will not increase the ratio in (5).

If we increase the radius of C (while fixing its center c) then $|p^* p^*_C|$ would increase but $|ap^*|$ and $|bp^*|$ remain unchanged. Thus, for the purpose of (5) we can assume that C has maximum possible radius which is δ. Then, for any point $q \in C$ it holds that $|qq_C| = |qc| + |cq_C| = |qc| + \delta$.



Let $\ell(a, b)$ be the line through a and b. If p^* lies in the same side of $\ell(a, b)$ as c does, then let $\overline{p^*}$ be the reflection of p^* with respect to $\ell(a, b)$; see the figure to the right. Notice that $\overline{p^*}$ also lies in lens L (which is not drawn in the figure). Moreover $|a\overline{p^*}| = |ap^*|$ and $|b\overline{p^*}| = |bp^*|$, but $|p^* p^*_C| \leqslant |\overline{p^*}\,\overline{p^*}_C|$ because p^* is closer to c than $\overline{p^*}$ to c. Thus, for the purpose of inequality (5), the point $\overline{p^*}$ achieves a smaller ratio than p^*. Therefore, we can assume that p^* lies in a different side of $\ell(a, b)$ than c does.

Let $\mathcal{L}$ denote the configuration that is the union of the lens L, the segment ab, and the point p^*. Notice that any translation, rotation, and reflection of $\mathcal{L}$ will not change $|ap^*|$ and $|bp^*|$. Consider the ray that is emanating from c and passing through p^*. Move $\mathcal{L}$ along this ray and stop as soon as one of a and b lies on the boundary of C. Assume that b is the point that lies on C. This translation can only increase $|p^* p^*_C|$, but not decrease. Now fix $\mathcal{L}$ at b and rotate it in the direction, that moves p^* away from c, until a also lies on the boundary of C. After rotation, one vertex of the lens L lies on c because the boundaries of $D(a, \delta)$ and $D(b, \delta)$ go through c; see Figure 4(b). Also, the lens L does not intersect the boundary of C because the distance of its other vertex to c is $2\sqrt{\delta^2 - (1/2)^2}$, which is strictly smaller than δ. The rotation can only increase $|p^* p^*_C|$, but not decrease. (Such a rotation moves p^*_C on the boundary of C, but that does not affect the argument because the value $|cp^*_C|$, which is equal to the radius of C, remains unchanged.) Therefore, above transformations do not increase the ratio in (5). After these transformations assume, without loss

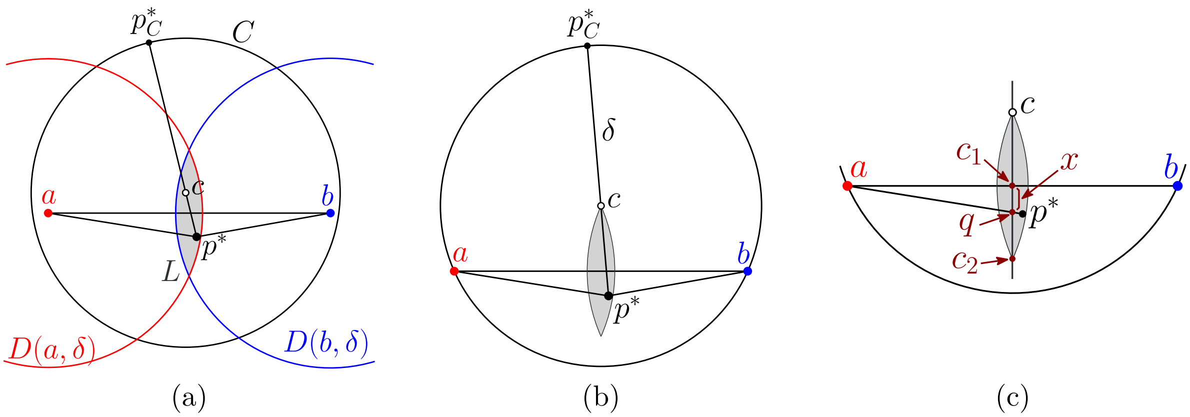


Figure 4: Illustration of (a) the lens L and the point p_C^* associated with p^*, (b) the configuration $\mathcal{L}$ after translation and rotation, (c) the points q, c_1, c_2.

of generality, that ab is horizontal, a is to the left of b, and c lies above ab. The current setting is depicted in Figure 4(b). Notice that $|p^*p_C^*| = |cp^*| + |cp_C^*| = |cp^*| + \delta$. Due to symmetry we may assume that p^* lies to the right side of the vertical line through c, and thus $|ap^*| \geqslant |bp^*|$, as shown in Figure 4(c). In the current setting, to achieve (5) it suffices to show that

$$\frac{|ap^*|}{|cp^*| + \delta} \geqslant \delta. \tag{6}$$

Let q be the intersection point of ap^* with the vertical line through c, as in Figure 4(c). Then $|ap^*| = |aq| + |qp^*|$ and $|cp^*| \leqslant |cq| + |qp^*|$. Thus,

$$\frac{|ap^*|}{|cp^*| + \delta} \geqslant \frac{|aq| + |qp^*|}{|cq| + |qp^*| + \delta} \geqslant \frac{|aq|}{|cq| + \delta},$$

where the second inequality is valid because we subtract the same amount $|qp^*|$ from the numerator and denominator of a fraction which is smaller than 1 (notice that $|aq| < |cq| + \delta$). Thus, to show (6) it suffices to show that

$$\frac{|aq|}{|cq| + \delta} \geqslant \delta. \tag{7}$$

Recall the definition of L and that its topmost point lies on the center c. Let c_1 be the intersection point of ab with the vertical line through c, and let c_2 be the lowest point of L; see Figure 4(c). Then $|cc_1| = |c_1c_2|$, $|ac| = |ac_2| = \delta$, and $|ac_1| = 1/2$. Notice that q lies on the segment c_1c_2, and $|cq| = |cc_1| + |qc_1|$. Denote the length $|qc_1|$ by x. Then $0 \leqslant x \leqslant |c_1c_2|$. Using the Pythagorean theorem we get $|aq| = \sqrt{x^2 + 1/4}$ and $|c_1c_2| = \sqrt{\delta^2 - 1/4}$. Thus we can write the ratio in (7) as a function f which depends only on x:

$$f(x) = \frac{|aq|}{|cq| + \delta} = \frac{\sqrt{x^2 + 1/4}}{x + \delta + \sqrt{\delta^2 - 1/4}},$$

where $x \in \left[0, \sqrt{\delta^2 - 1/4}\right]$. The function $f(x)$ is monotonically decreasing on this interval of x and thus its minimum value is attained at $\sqrt{\delta^2 - 1/4}$. Plugging this into f we get $f\left(\sqrt{\delta^2 - 1/4}\right) = \frac{\sqrt{7}-1}{3} = \delta$. This verifies (7) and finishes the proof of the lemma. $\square$

The cases considered in Lemmas 1, 3, and 4 ensure that the length of one of S_1, S_2, S_3, and D is at least $\delta \cdot \text{len}(T^*)$. This concludes our analysis and finishes the proof of Theorem 1.

3.2 Inclusion of bichromatic diameter

Here we show that the approximation ratio of an algorithm, that always includes a bichromatic diametral pair in its solution, cannot be larger than 0.5.

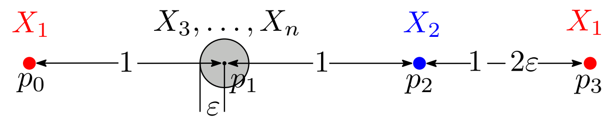


Figure 5: Illustration of the upper bound 0.5 for inclusion of a bichromatic diametral pair.

We introduce an input instance with n neighborhoods. Consider four points $p_0 = (0, 0)$, $p_1 = (1, 0)$, $p_2 = (2, 0)$, and $p_3 = (3-2\varepsilon, 0)$ for arbitrary small $\varepsilon > 0$, e.g. $\varepsilon = 1/n$. Our input consists of neighborhoods $X_1, \ldots, X_n$ where $X_1 = \{p_0, p_3\}$, $X_2 = \{p_2\}$, and each of $X_3, \ldots, X_n$ has exactly one point that is placed at distance at most ε from p_1; see Figure 5. In this setting, (p_0, p_2) is the unique bichromatic diametral pair. Consider any tree T that contains the bichromatic diameter p_0p_2 (this means that p_3 is not in T). Any edge of T incident to $X_3, \ldots, X_4$ has length at most $1+\varepsilon$. Therefore $\text{len}(T) \leqslant 2 + (1+\varepsilon)(n-2) < n+1$. Now consider another tree T' that does not contain p_0p_2 but connects each of $X_2, \ldots, X_n$ to p_3. The length of T' is at least $(1-2\varepsilon) + (2-3\varepsilon)(n-2) > 2n-6$. Then, the ratio

$$\frac{\text{len}(T)}{\text{len}(T')} < \frac{n+1}{2n-6}$$

tends to $1/2$ in the limit. This establishes the upper bound 0.5 on the approximation ratio.

4 Conclusions

A natural open problem is to further improve the approximation ratio for the Max-ST-NB problem. We believe that our algorithm has better approximation guarantee, however this requires more detailed analysis. We obtained the ratio of 0.548 by analyzing the stars S_1, S_2, S_3 and the double-star D separately. One might be able to improve the ratio by analyzing the stars and the double-star together and then taking the longest one.

References

[1] P. K. Agarwal, J. Matousek, and S. Suri. Farthest neighbors, maximum spanning trees and related problems in higher dimensions. *Comput. Geom.*, 1:189–201, 1991.

[2] N. Alon, S. Rajagopalan, and S. Suri. Long non-crossing configurations in the plane. *Fundam. Inform.*, 22(4):385–394, 1995. Also in *SoCG* 1993.

[3] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discret. Appl. Math.*, 55(3):197–218, 1994.

[4] T. Asano, B. K. Bhattacharya, J. M. Keil, and F. F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the 4th Annual Symposium on Computational Geometry (SoCG)*, pages 252–257, 1988.

[5] M. Bern and D. Eppstein. Approximation algorithms for geometric problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, page 296–345. PWS Publishing, 1996.

[6] B. K. Bhattacharya and G. T. Toussaint. Efficient algorithms for computing the maximum distance between two finite planar sets. *J. Algorithms*, 4(2):121–136, 1983.

[7] A. Biniaz, P. Bose, D. Eppstein, A. Maheshwari, P. Morin, and M. Smid. Spanning trees in multipartite geometric graphs. *Algorithmica*, 80(11):3177–3191, 2018.

[8] V. Blanco, E. Fernández, and J. Puerto. Minimum spanning trees with neighborhoods: Mathematical programming formulations and solution methods. *Eur. J. Oper. Res.*, 262(3):863–878, 2017.

[9] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21(3):579–597, 1996.

[10] K. Chen and A. Dumitrescu. On the longest spanning tree with neighborhoods. *Discrete Mathematics, Algorithms and Applications*, 12(5), 2020. Also in FAW'18.

[11] M. de Berg, O. Cheong, M. J. van Kreveld, and M. H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition.* Springer, 2008.

[12] M. de Berg, J. Gudmundsson, M. J. Katz, C. Levcopoulos, M. H. Overmars, and A. F. van der Stappen. TSP with neighborhoods of varying size. *J. Algorithms*, 57(1):22–36, 2005. Also in *ESA* 2002.

[13] R. Dorrigiv, R. Fraser, M. He, S. Kamali, A. Kawamura, A. López-Ortiz, and D. Seco. On minimum- and maximum-weight minimum spanning trees with neighborhoods. *Theory Comput. Syst.*, 56(1):220–250, 2015.

[14] A. Dumitrescu and C. D. Tóth. Long non-crossing configurations in the plane. *Discret. Comput. Geom.*, 44(4):727–752, 2010. Also in *STACS* 2010.

[15] G. A. Galperin. A concept of the mass center of a system of material points in the constant curvature spaces. *Communications in Mathematical Physics*, 154(1):63–84, 1993.

[16] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing* (*STOC*), pages 585–594, 2003.

[17] M. Löffler. *Data Imprecision in Computational Geometry.* Phd thesis, Utrecht University, 2009.

[18] M. Löffler and M. J. van Kreveld. Largest and smallest convex hulls for imprecise points. *Algorithmica*, 56(2):235–269, 2010.

[19] N. Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983. Also in *FOCS 1982*.

[20] J. S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (*SODA*), pages 11–18, 2007.

[21] J. S. B. Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Proceedings of the 26th ACM Symposium on Computational Geometry* (*SoCG*), pages 183–191, 2010.

[22] C. L. Monma, M. Paterson, S. Suri, and F. F. Yao. Computing Euclidean maximum spanning trees. *Algorithmica*, 5(3):407–419, 1990.

[23] M. J. van Kreveld and M. Löffler. Approximating largest convex hulls for imprecise points. *J. Discrete Algorithms*, 6(4):583–594, 2008.

[24] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In *Maurer H. (eds) New Results and New Trends in Computer Science*, pages 359–370. LNCS 555, 1991.

[25] Y. Yang, M. Lin, J. Xu, and Y. Xie. Minimum spanning tree with neighborhoods. In *Proceedings of the 3rd International ConferenceAlgorithmic Aspects in Information and Management* (*AAIM*), pages 306–316, 2007.