



# Minimum Plane Bichromatic Spanning Trees

HUGO A. AKITAYA, University of Massachusetts Lowell, USA

AHMAD BINIAZ, University of Windsor, Canada

ERIK D. DEMAINE, Massachusetts Institute of Technology, USA

LINDA KLEIST, Universität Potsdam, Germany

FREDERICK STOCK, University of Massachusetts Lowell, USA

CSABA D. TÓTH, California State University Northridge, USA and Tufts University, USA

For a set of red and blue points in the plane, a *minimum bichromatic spanning tree* (MinBST) is a shortest spanning tree of the points such that every edge has a red and a blue endpoint. A MinBST can be computed in  $O(n \log n)$  time where  $n$  is the number of points. In contrast to the standard Euclidean MST, which is always *plane* (noncrossing), a MinBST may have edges that cross each other. However, we prove that a MinBST is quasi-plane, that is, it does not contain three pairwise crossing edges, and we determine the maximum number of crossings.

Moreover, we study the problem of finding a *minimum plane bichromatic spanning tree* (MinPBST) which is a shortest bichromatic spanning tree with pairwise noncrossing edges. This problem is known to be NP-hard. The previous best approximation algorithm, due to Borgelt et al. (2009), has a ratio of  $O(\sqrt{n})$ . It is also known that the optimum solution can be computed in polynomial time in some special cases, for instance, when the points are in convex position, collinear, semi-collinear, or when one color class has constant size. We present an  $O(\log n)$ -factor approximation algorithm for the general case.

CCS Concepts: • **Theory of computation** → **Approximation algorithms analysis**; **Computational geometry**; • **Mathematics of computing** → *Combinatoric problems*.

Additional Key Words and Phrases: Bichromatic Spanning Tree, Minimum Spanning Tree, Plane Tree<sup>1</sup>

## 1 INTRODUCTION

Computing a minimum spanning tree (MST) in a graph is a well-studied problem. There exist many algorithms for this problem, among which one can mention the celebrated Kruskal's algorithm [37], Prim's algorithm [41], and Borůvka's algorithm [22]. The running time of these algorithms depends on the number of vertices and edges of the input graph. For complete geometric graphs, where the vertices are points in the plane, these algorithms can be applied in such a way that their running time depends only on the number of vertices.

<sup>1</sup>A preliminary version of this paper appeared in the *Proceedings of the 35th International Symposium on Algorithms and Computation (ISAAC)*, volume 322 of *LIPIcs*, 4:1–4:14, Schloss Dagstuhl, 2024. <http://dx.doi.org/10.4230/LIPIcs.ISAAC.2024.4>.

Authors' Contact Information: Hugo A. Akitaya, Miner School of Computer & Information Sciences, University of Massachusetts Lowell, Lowell, USA, [hugo\\_akitaya@uml.edu](mailto:hugo_akitaya@uml.edu); Ahmad Biniaz, School of Computer Science, University of Windsor, Windsor, Canada, [abiniaz@uwindsor.ca](mailto:abiniaz@uwindsor.ca); Erik D. Demaine, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA, [edemaine@mit.edu](mailto:edemaine@mit.edu); Linda Kleist, Department of Computer Science, Universität Potsdam, Potsdam, Germany, [kleist@cs.uni-potsdam.de](mailto:kleist@cs.uni-potsdam.de); Frederick Stock, Miner School of Computer & Information Sciences, University of Massachusetts Lowell, Lowell, USA, [frederick\\_stock@student.uml.edu](mailto:frederick_stock@student.uml.edu); Csaba D. Tóth, Department of Mathematics, California State University Northridge, Los Angeles, CA, USA, Department of Computer Science and Tufts University, Medford, MA, USA, [csaba.toth@csun.edu](mailto:csaba.toth@csun.edu).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, or post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2025 Copyright held by the owner/author(s).

ACM 1549-6333/2025/7-ART

<https://doi.org/10.1145/3747591>

For a set  $S$  of  $n$  points in the plane a Euclidean MST (i.e., an MST of the complete graph on  $S$  with straight-line edges and Euclidean edge weights) can be computed in  $O(n \log n)$  time. When the points of  $S$  are colored by two colors, say red and blue, and every edge is required to have a red and a blue endpoint, then a spanning tree is referred to as a *bichromatic spanning tree*. A *minimum bichromatic spanning tree* (MinBST) is a bichromatic spanning tree of minimum total edge length. A MinBST on  $S$  can be computed in  $O(n \log n)$  time [18]. When the points are collinear (all lie on a straight line) and are given in sorted order along the line this problem can be solved in linear time [15].

We say that two line segments *cross* if they share an interior point; this configuration is called a *crossing*. A tree is called *plane* if its edges are pairwise noncrossing. The standard Euclidean MST is always plane. This property is ensured by the triangle inequality, because the tree can be made shorter by replacing any two crossing edges with two noncrossing edges. The noncrossing property does not necessarily hold for a MinBST, see Figure 1 for an example. Two crossing edges in this example cannot be replaced with two noncrossing edges because, otherwise, we would either introduce monochromatic edges (that connect points of the same color) or disconnect the tree into two components.

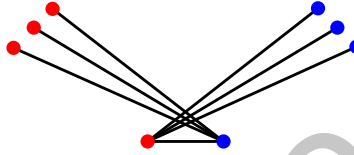


Fig. 1. A bicolored point set and its minimum bichromatic spanning tree (MinBST).

Edge crossings in geometric graphs are usually undesirable as they could lead to unwanted situations such as collisions in motion planning, inconsistency in VLSI layout, and interference in wireless networks. They are also undesirable in the context of graph drawing and network visualization. Therefore, it is natural to ask for a *minimum plane bichromatic spanning tree* (MinPBST), a bichromatic spanning tree that is noncrossing and has minimum total edge length. Borgelt et al. [21] proved that the problem of finding a MinPBST is NP-hard. They also present a polynomial-time approximation algorithm with approximation factor  $O(\sqrt{n})$ .

In this paper we study the MinBST and MinPBST problems from combinatorial and computational points of view. First we present an approximation algorithm, with a better factor, for the MinPBST problem. Then we prove some interesting structural properties of the MinBST.

### 1.1 Related work

Problems related to bichromatic objects (such as points and lines) have been actively studied in computational geometry, for instance, the problems related to bichromatic intersection [4, 24, 25, 38], bichromatic separation [9, 11, 14, 16, 27], and noncrossing bichromatic connection [1, 2, 17, 19, 21, 32, 34, 36]. We refer the interested reader to the survey by Kaneko and Kano [35].

The  $O(\sqrt{n})$ -approximation algorithm of Borgelt et al. [21] for the MinPBST problem lays a  $(\sqrt{n} \times \sqrt{n})$ -grid over the points, then identifies a subset of grid cells as *core* regions and computes their Voronoi diagram, then builds a tree inside each Voronoi cell, and finally combines the trees.

Let  $\rho_n$  be the supremum ratio of the length of a MinPBST to the length of a MinBST over all sets of  $n$  bichromatic points. Grantson et al. [29] show that  $3/2 \leq \rho_n \leq n$  for all  $n \geq 4$ ; and ask whether the upper bound can be improved. It is easily seen from the algorithm of Borgelt et al. [21] that  $\rho_n \leq O(\sqrt{n})$  because the planarity of the optimal solution is not used in the analysis of the approximation ratio—indeed the analysis would work even with respect to the MinBST.

Some special cases of the MinPBST problem can be solved to optimality in polynomial time. For instance, the problem can be solved in  $O(n^2)$  time when points are collinear [15], in  $O(n^3)$  time when points are in convex position [21], in  $O(n^5)$  time when points are semi-collinear (points in one color class are on a line and all other points are on one side of the line) [17], and in  $n^{O(k^5)}$  time when one color class has  $k$  points for some constant  $k$  [21].

One might wonder if a greedy strategy could achieve a better approximation ratio. A modified version of Kruskal's algorithm, that successively adds a shortest bichromatic edge that creates neither a cycle nor a crossing, is referred to as the *greedy algorithm* [21, 29]. This algorithm, as noted in [21, Figure 1], does not always return a planar bichromatic tree (it does not always terminate: there may be a point of one color that cannot see any point of the opposite color).

Abu-Affash et al. [2] studied the *bottleneck* version of the plane bichromatic spanning tree problem where the goal is to minimize the length of the longest edge. They prove that this problem is NP-hard, and present an  $8\sqrt{2}$ -approximation algorithm.

## 1.2 Quasi-planarity

Quasi-planarity is a measure of the proximity of an (abstract or geometric) graph to planarity. For an integer  $k \geq 2$ , a graph is called *k-quasi-planar* if it can be drawn in the plane such that no  $k$  edges pairwise cross. By this definition, a planar graph is 2-quasi-planar. A 3-quasi-planar graph is also called *quasi-planar*. Problems on  $k$ -quasi-planarity are closely related to Turán-type problems on the intersection graph of line segments in the plane [5, 10, 23, 28]. They are also related to the size of *crossing families* (pairwise crossing edges) determined by points in the plane [12, 40]. Perhaps a most notable question on quasi-planarity is a conjecture by Pach, Shahrokhi, and Szegedy [39] that for any fixed integer  $k \geq 3$ , there exists a constant  $c_k$  such that every  $n$ -vertex  $k$ -quasi-planar graph has at most  $c_k n$  edges. This conjecture has been verified for  $k = 3$  [5] and  $k = 4$  [3].

A drawing of a graph is called *k-quasi-plane* if no  $k$  edges in the drawing pairwise cross, and a drawing is *quasi-plane* if it is 3-quasi-plane. For example, the drawing of a tree in Figure 1 is quasi-plane. This concept plays an important role in decompositions of geometric graphs: Aichholzer et al. [6] showed recently that the complete geometric graph on  $2n$  points in the plane can always be decomposed into  $n$  quasi-plane spanning trees (but not necessarily into  $n$  plane spanning trees).

## 1.3 Our contributions

In Section 2 we present a randomized approximation algorithm with factor  $O(\log n)$  for the MinPBST problem. Our algorithm computes a randomly shifted quadtree on the points, and then builds a planar bichromatic tree in a bottom-up fashion from the leaves of the quadtree towards the root. We then derandomize the algorithm by discretizing the random shifts. Our weight analysis shows that  $|\text{MinBST}(S)| \leq |\text{MinPBST}(S)| \leq O(\log n) \cdot |\text{MinBST}(S)|$  for every set  $S$  of  $n$  bichromatic points, which implies that  $\rho_n = O(\log n)$ .

In Section 3.1 we prove that every MinBST is quasi-plane, i.e., no three edges pairwise cross in its inherited drawing (determined by the point set). In a sense, this means that MinBST is not far from being a plane graph. In Section 3.2 we determine the maximum number of crossings in a MinBST. We conclude with a list of open problems in Section 4.

## 2 APPROXIMATION ALGORITHM FOR MINPBST

In this section we first present a randomized approximation algorithm for the MinPBST problem. Then we show how to derandomize the algorithm at the expense of increasing the running time by a quadratic factor. The following theorem summarizes our result in this section. Throughout this section we consider point sets in the plane that are in general position, that is, no three points lie on a straight line.

**THEOREM 2.1.** *There is a randomized algorithm that, given a set of  $n$  red and blue points in the plane in general position, returns a plane bichromatic spanning tree of expected weight at most  $O(\log n)$  times the optimum, and runs in  $O(n \log^2 n)$  time. The algorithm can be derandomized by increasing the running time by a factor of  $O(n^2)$ .*

Let  $S$  be a set of  $n$  red and blue points in the plane. To simplify our arguments we assume that  $n$  is a power of 2. Let  $\text{OPT}$  denote the length of a minimum bichromatic spanning tree on  $S$  (and note that  $\text{OPT}$  is an obvious lower bound for the length of a minimum *plane* bichromatic spanning tree on  $S$ ). Our algorithm computes a plane bichromatic spanning tree of expected length  $O(\log n) \cdot \text{OPT}$ .

## 2.1 Preliminaries for the algorithm

The following folklore lemma, though very simple, plays an important role in our construction.

**LEMMA 2.2.** *Every set of  $n$  red and blue points in general position in the plane, containing at least one red and at least one blue point, admits a plane bichromatic spanning tree. Such a tree can be computed in  $O(n \log n)$  time.*

A proof of Lemma 2.2 can be found in [17]. Essentially such a tree can be constructed by connecting an arbitrary red point to all blue points (this partitions the plane into cones) and then connecting red points in each cone to a blue point on its boundary.

For a connected geometric graph  $G$  and a point  $q$  in the plane, we say that  $q$  *sees* an edge  $(a, b)$  of  $G$  if the interior of the triangle  $\triangle qab$  is disjoint from vertices and edges of  $G$ . In other words, the entirety of the edge  $(a, b)$  is visible from  $q$ . The following lemma (that is implied from [33, Lemma 2.1]) also plays an important role in our construction.

**LEMMA 2.3.** *Let  $G$  be a connected plane geometric graph with  $n$  vertices and  $q$  be a point outside the convex hull of the vertices of  $G$ . Then  $q$  sees an edge of  $G$ . Such an edge can be found in  $O(n \log n)$  time.*

Note that the condition that  $q$  lies outside of the convex hull of  $G$  is necessary, as otherwise  $q$  may not see any edge of  $G$  entirely. The application of this lemma to our algorithm is that if  $G$  is properly colored and a vertex sees an edge  $(a, b)$ , then  $(q, a)$  or  $(q, b)$  is bichromatic and does not cross any edges of  $G$ . This idea was previously used in [31–33].

## 2.2 The algorithm

After a suitable scaling, we may assume that the smallest axis-aligned square containing  $S$  has side length 1. After a suitable translation, we may assume that the lower left corner of this square is the point  $(1, 1)$ ; and so its top right corner is  $(2, 2)$  as in Figure 2(a). Observe that

$$\text{OPT} \geq 1.$$

Our algorithm employs a randomly shifted quadtree as in Arora’s PTAS for the Euclidean TSP [13]. Let  $Q$  be a  $2 \times 2$  axis-aligned square whose lower left corner is the origin— $Q$  contains all points of  $S$ . Subdivide  $Q$  into four congruent squares, and recurse until  $Q$  is subdivided into squares of side length  $1/n$  as in Figure 2(a). The depth of this recursion is  $1 + \log n$ . For the purpose of shifting, pick two real numbers  $x$  and  $y$  in the interval  $[0, 1]$  uniformly at random. Then translate  $Q$  such that its lower left corner becomes  $(x, y)$  as in Figure 2(b). This process is called a *random shift*. The points of  $S$  remain in  $Q$  after the shift. We obtain a quadtree subdivision of  $S$  of depth at most  $1 + \log n$  with respect to the subdivision of  $Q$ , i.e., the lines of the quadtree subdivision are chosen from the lines of the subdivision of  $Q$ ; see Figure 2(c). The resulting quadtree is called (randomly) *shifted quadtree*. We stop the recursive subdivision at squares that have size  $1/n \times 1/n$  or that are *empty* (disjoint from  $S$ ) or *monochromatic* (have points of only one color). Therefore a leaf-square of the quadtree may contain more than one point of  $S$ .

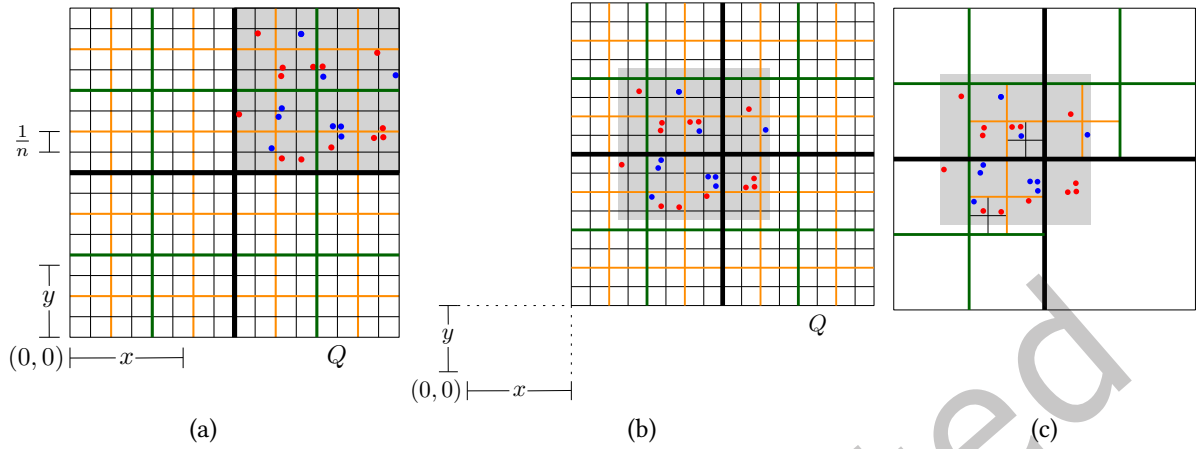


Fig. 2. (a) Shaded square contains  $S$ . (b) Translated subdivision of  $Q$ . (c) Randomly shifted quadtree on points of  $S$  with respect to the subdivision of  $Q$ .

At the root level (which we may consider as level  $-1$ ) we have  $Q$  which has size  $2 \times 2$ . At each recursive level  $i = 0, 1, \dots, \log n$  we have squares of size  $1/2^i \times 1/2^i$  (Level 0 stands for the first time that we subdivide  $Q$ ). Thus at level 0 we have four squares of size  $1 \times 1$ , and at level  $\log n$  we have squares of size  $1/n \times 1/n$ . Our strategy is to use the shifted quadtree and compute an approximate solution in a bottom-up fashion from the leaves towards the root. For each square that is *bichromatic* (contains points of both colors) we will find a plane bichromatic spanning tree of its points. For monochromatic squares, we do not do anything. At the root level, we have  $Q$  which contains  $S$  and is bichromatic, so we will get an approximate plane bichromatic spanning tree of  $S$ .

At level  $i = \log n$ , we have squares of size  $1/n \times 1/n$ , and thus the length of any edge in such squares is at most  $\sqrt{2}/n$ . For each bichromatic square we compute a plane bichromatic tree arbitrarily, for instance by Lemma 2.2. For monochromatic squares we do nothing. The spanning trees, for all the squares of size  $1/n \times 1/n$ , have less than  $n$  edges in total. Hence the total length of all these trees is at most

$$\frac{\sqrt{2}}{n} \cdot n = \sqrt{2} \leq \sqrt{2} \cdot \text{OPT}.$$

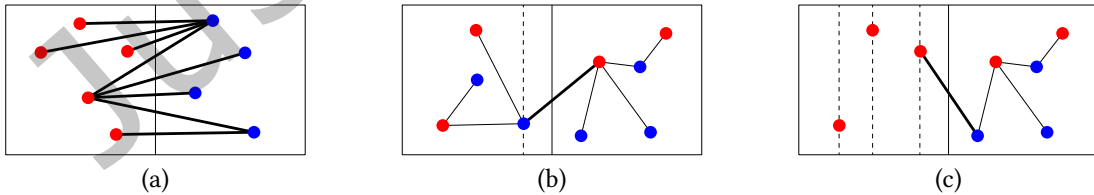


Fig. 3. Merging two squares in the same row: (a) case 1, (b) case 2, and (c) case 3.

At each level  $i < \log n$ , we have already solved the problem within squares of level  $i + 1$ . Each square at level  $i$  has size  $1/2^i \times 1/2^i$ . If the square is monochromatic we do nothing. If the square is bichromatic, then it consists of four squares at level  $i + 1$ . We merge the solutions of the four squares to obtain a solution for the level  $i$  square as

follows. First we merge the solutions in adjacent squares in the same row (we have two such pairs), and then merge the two solutions in the two rows. Thus each merge is performed on two solutions that are separated by a (vertical or horizontal) line. For the merge we apply one of the following cases:

- (1) If each merge party is monochromatic but their union is bichromatic, then we construct an arbitrary plane bichromatic tree on the union, for example by Lemma 2.2. See Figure 3(a).
- (2) If both merge parties are bichromatic (and hence are trees), then we take a point in one tree that is closest to the square (or rectangle) containing the other tree and merge them using Lemma 2.3. See Figure 3(b).
- (3) If one part is bichromatic (a tree) and the other is monochromatic, then we first sort the points in the monochromatic square (or rectangle) in increasing order according to their distance to the bichromatic square (or rectangle). Then we merge the points (one at a time) with the current bichromatic tree by using Lemma 2.3. See Figure 3(c).

In each case, the merge produces a plane bichromatic tree in the level- $i$  square. We process all squares in a bottom-up traversal of the quadtree. In the end, after processing level  $-1$ , we get a plane bichromatic spanning tree for points of  $S$  in square  $Q$ . Denote this tree by  $T$ .

### 2.3 Weight analysis

We start by introducing an alternative measurement for the length of the optimal tree, i.e., MinBST. This new measurement, denoted by  $\text{OPT}'$ , will be used to bound the length of our tree  $T$ . We say that a quadtree line is at level  $i$  if it contains a side of some level- $i$  square. A side of a level- $i$  square (that is not a leaf) gets subdivided to yield sides of two squares at level  $i + 1$ . Thus any quadtree line at level  $i$  is also at levels  $i + 1, i + 2, \dots, \log n$ .

For each level  $i \in \{0, 1, \dots, \log n\}$  we define a parameter  $\text{OPT}'_i$ . Let  $T_{\text{opt}}$  be a MinBST for  $S$ . For each edge  $e$  of  $T_{\text{opt}}$ , define the indicator variable

$$X_i(e) = \begin{cases} 1 & \text{if } e \text{ intersects the boundary of a level-}i \text{ square of the quadtree,} \\ 0 & \text{otherwise.} \end{cases}$$

Now let

$$\text{OPT}'_i = \sum_{e \in E(T_{\text{opt}})} \frac{1}{2^i} X_i(e),$$

that is, a weighted sum of  $X_i(e)$  over all edges of  $T_{\text{opt}}$ , and let

$$\text{OPT}' = \sum_{i=0}^{\log n} \text{OPT}'_i.$$

The new measure  $\text{OPT}'$  can be arbitrarily large compared to  $\text{OPT}$  in the worst case. For example if the optimal solution is a path consisting of  $n - 1$  edges of small length say  $2/n$ , such that each of them intersects the vertical line at level 0, then  $\text{OPT}$  is roughly 2 but  $\text{OPT}'$  is at least  $n - 1$ . However, using the random shift at the beginning of our algorithm, we can show that the expected value of  $\text{OPT}'$  is not very large compared to  $\text{OPT}$ .

LEMMA 2.4.  $\mathbb{E}[\text{OPT}'] \leq \sqrt{2}(1 + \log n) \cdot \text{OPT}$ .

PROOF. Consider any edge  $e$  of length  $\ell$  in MinBST. Let  $\ell_x$  denote the  $x$ -span of  $e$ , i.e., the difference of the  $x$ -coordinates of the two endpoints of  $e$ , and let  $\ell_y$  be the  $y$ -span of  $e$ . Observe that  $\ell_x + \ell_y \leq \sqrt{2}\ell$ . The probability that  $e$  intersects a vertical line at any level  $i$  is

$$\Pr(e \text{ intersects a vertical line at level } i) \leq \ell_x 2^i,$$



because there are  $2^i$  uniformly spaced but randomly shifted vertical lines at level  $i$ . Combining this with the probability of intersecting horizontal lines, the union bound yields

$$\Pr(X_i(e)) = \Pr(e \text{ intersects a line at level } i) \leq (\ell_x + \ell_y)2^i \leq \sqrt{2}\ell 2^i.$$

If  $e$  intersects a line at level  $i$ , then  $1/2^i$  is added to  $\text{OPT}'_i$ , otherwise nothing is added. Thus the expected added value for  $e$  to  $\text{OPT}'_i$  is

$$\mathbb{E} \left[ \frac{1}{2^i} X_i(e) \right] = \frac{1}{2^i} \cdot \Pr(X_i(e)) \leq \frac{1}{2^i} \cdot \sqrt{2}\ell 2^i = \sqrt{2}\ell.$$

Summation over all edges of the optimal tree  $T_{\text{opt}}$  gives  $\mathbb{E}[\text{OPT}'_i] \leq \sqrt{2} \cdot \text{OPT}$ , and summation over all levels yields  $\mathbb{E}[\text{OPT}'] \leq \sqrt{2}(1 + \log n) \cdot \text{OPT}$ .  $\square$

We already know that the total length of trees constructed in level  $\log n$  is at most  $\sqrt{2} \cdot \text{OPT}$ . Let  $E_i$  be the set of all edges that were added to  $T$  at level  $i < \log n$  in the bottom-up construction. We establish a correspondence between the edges in  $E_i$  and the values added to  $\text{OPT}'$ . The edges of  $E_i$  were added by cases (1), (2), and (3). We consider each case separately.

- (1) Assume that the union of merge parties has  $k$  points. Then we add  $k - 1$  edges of length at most  $\sqrt{2} \cdot 1/2^i$  to  $E_i$ . For these points,  $T_{\text{opt}}$  also needs to have at least  $k - 1$  connections that intersect the boundaries of the squares involved in the merge, which have side length at least  $1/2^{i+1}$ . For each such edge we have added a value of  $1/2^{i+1}$  to  $\text{OPT}'_{i+1}$ .
- (2) To merge the two trees, we added just one edge of length at most  $\sqrt{2} \cdot 1/2^i$  to  $E_i$ . For each merge party,  $T_{\text{opt}}$  needs at least one edge that crosses the boundary of its rectangle. These edges, however, need not be distinct (e.g., an edge can cross the boundary between the two rectangles). In any case,  $T_{\text{opt}}$  needs at least one edge that crosses the boundary of one of the two rectangles, for which we have added at least  $1/2^{i+1}$  to  $\text{OPT}'_{i+1}$ .
- (3) Assume that the monochromatic party has  $k$  points. Thus we added  $k$  edges of length at most  $\sqrt{2} \cdot 1/2^i$  to  $E_i$ . Again,  $T_{\text{opt}}$  needs at least  $k$  edges that cross the boundary of the squares involved in the merge; and for each such edge we have added at least  $1/2^{i+1}$  to  $\text{OPT}'_{i+1}$ .

Therefore the total length of all edges that were added to  $T$  at level  $i$  is at most  $|E_i| \cdot \sqrt{2}/2^i$ . Analogously, the total value that has been added to  $\text{OPT}'_{i+1}$  is at least  $1/2 \cdot |E_i| \cdot 1/2^{i+1} = |E_i| \cdot 1/2^{i+2}$ . The multiplicative factor  $1/2$  comes from the fact that the two endpoints of an edge of the optimal tree could be involved in two separate merge operations. By summing the length of edges added to  $T$  in all levels and considering Lemma 2.4 we get

$$\begin{aligned} \mathbb{E}[|T|] &\leq \sqrt{2} \cdot \text{OPT} + \mathbb{E} \left[ \sum_{i=0}^{\log n - 1} |E_i| \frac{\sqrt{2}}{2^i} \right] \leq \sqrt{2} \cdot \text{OPT} + 4\sqrt{2} \cdot \mathbb{E} \left[ \sum_{i=0}^{\log n - 1} |E_i| \frac{1}{2^{i+2}} \right] \\ &\leq \sqrt{2} \cdot \text{OPT} + 4\sqrt{2} \cdot \mathbb{E} \left[ \sum_{i=0}^{\log n - 1} \text{OPT}'_{i+1} \right] \leq \sqrt{2} \cdot \text{OPT} + 4\sqrt{2} \cdot \mathbb{E}[\text{OPT}'] \\ &\leq \sqrt{2} \cdot \text{OPT} + 4\sqrt{2} \cdot \sqrt{2}(1 + \log n) \cdot \text{OPT} = O(\log n) \cdot \text{OPT}. \end{aligned}$$

## 2.4 Running time

The initial shifted quadtree has depth  $O(\log n)$  and has  $O(n)$  leaves. Thus it can be computed in  $O(n \log n)$  time by a divide-and-conquer sorting-based algorithm [26]. From a result of [17] (cf. Lemma 2.2) it follows that the bichromatic trees at the leaves of the quadtree can be computed in total  $O(n \log n)$  time. From a result of [30] and

[33] (cf. Lemma 2.3) it follows that the total merge time in each level of the quadtree is  $O(n \log n)$ . Summing over all levels, the running time of our randomized algorithm is  $O(n \log^2 n)$ .

## 2.5 Derandomization

In the algorithm of Section 2.2, we shifted the  $2 \times 2$  square  $Q$  by a real vector  $(x, y)$ , where  $x$  and  $y$  are chosen independently and uniformly at random from the interval  $[0, 1]$ . We now *discretize* the random shift, and choose  $x$  and  $y$  independently and uniformly at random from the finite set  $\{0, 1/n, 2/n, \dots, (n-1)/n\}$ . We call this process the *discrete random shift*. We show that the proof of Lemma 2.4 can be adapted under this random experiment with a larger constant coefficient. Therefore we can derandomize the algorithm by trying all random choices of  $(x, y)$  for the shift and return the shortest tree over all choices. This increases the running time by a factor of  $O(n^2)$ .

LEMMA 2.5. *Under the discrete random shift, we have  $\mathbb{E}[\text{OPT}'] \leq (\sqrt{2} + 2)(1 + \log n) \cdot \text{OPT}$ .*

PROOF. Consider any edge  $e = ab$  of length  $\ell$  in the optimal tree  $T_{\text{opt}}$ . The two endpoints of  $e$  are points  $a = (a_x, a_y)$  and  $b = (b_x, b_y)$ . Denote the orthogonal projection of  $e$  to the  $x$ - and  $y$ -axes by  $e_x$  and  $e_y$ , respectively, and observe that  $e_x = [\min\{a_x, b_x\}, \max\{a_x, b_x\}]$  and  $e_y = [\min\{a_y, b_y\}, \max\{a_y, b_y\}]$ . We discretize these intervals as follows. Replace each endpoint of the interval  $e_x$  with the closest rationals of the form  $k/n + 1/2n$ , and let  $e'_x$  be the resulting interval; similarly we obtain  $e'_y$  from  $e_y$ . Observe that  $e_x$  intersects a vertical line of the form  $x = a/n$ ,  $a \in \mathbb{Z}$ , if and only if  $e'_x$  does. Let  $\ell'_x$  and  $\ell'_y$  denote the lengths of intervals  $e'_x$  and  $e'_y$ , respectively. By construction, we have  $\ell'_x \leq \ell_x + 1/n$  and  $\ell'_y \leq \ell_y + 1/n$ . Consequently, the probability that  $e$  intersects a vertical line at any level  $i$  is

$$\begin{aligned} \Pr(e \text{ intersects a vertical line at level } i) &= \Pr(e'_x \text{ intersects a vertical line at level } i) \\ &\leq \ell'_x 2^i \leq (\ell_x + 1/n) 2^i, \end{aligned}$$

because there are  $2^i$  uniformly spaced but randomly shifted vertical lines at level  $i$ . Combining this with the probability of intersecting horizontal lines, the union bound yields

$$\Pr(X_i(e)) = \Pr(e \text{ intersects a line at level } i) \leq (\ell_x + \ell_y + 2/n) 2^i \leq (\sqrt{2}\ell + 2/n) 2^i.$$

The expected added value for  $e$  to  $\text{OPT}'_i$  is

$$\mathbb{E} \left[ \frac{1}{2^i} X_i(e) \right] = \frac{1}{2^i} \cdot \Pr(X_i(e)) \leq \frac{1}{2^i} \cdot \left( \sqrt{2}\ell + \frac{2}{n} \right) 2^i = \sqrt{2}\ell + \frac{2}{n}.$$

Summation over all edges of the optimal tree  $T_{\text{opt}}$  gives  $\mathbb{E}[\text{OPT}'_i] \leq \sqrt{2} \cdot \text{OPT} + 2 \leq (\sqrt{2} + 2) \cdot \text{OPT}$ , and summation over all levels yields  $\mathbb{E}[\text{OPT}'] \leq (\sqrt{2} + 2)(1 + \log n) \cdot \text{OPT}$ .  $\square$

## 2.6 Generalization to more colors

Our approximation algorithm for the MinPBST (minimum plane two-colored spanning tree) can be simply generalized to more colors. In this general setting, we are given a colorful point set  $S$  and we want to find a spanning tree on  $S$  with properly colored edges, i.e., the two endpoints of every edge should be of different colors. The same quadtree approach would give a plane properly-colored spanning tree. The analysis and the approximation ratio would be the same mainly because whenever we introduce some edges to merge two squares, the optimal solution must have the same number of edges that cross the boundaries of the squares. Also the running time remains the same because Lemma 2.2 and Lemma 2.3 carry over to multicolored point sets.



### 3 CROSSING PATTERNS IN MINBST

In this section, we prove that every MinBST is quasi-plane for every 2-colored point set in general position (Section 3.1), and then use this result to determine the maximum number of crossings in a MinBST for a set of  $n$  bichromatic points in the plane (Section 3.2).

#### 3.1 Quasi-planarity

Let  $S$  be a set of red and blue points in the plane. To differentiate between the points we denote the red points by  $r_1, r_2, \dots$  and the blue points by  $b_1, b_2, \dots$ . Let  $T$  be a MinBST for  $S$ . For two distinct edges  $e_1$  and  $e_2$  of  $T$  we denote the unique shortest path between  $e_1$  and  $e_2$  in  $T$  by  $\delta(e_1, e_2)$ . This path contains exactly one endpoint of  $e_1$  and one endpoint of  $e_2$ .

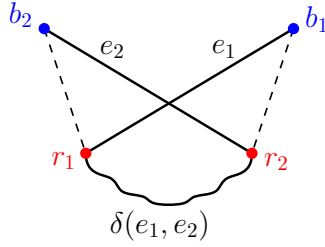


Fig. 4. Illustration of the proof of Lemma 3.1. Uncrossing a pair of crossing edges.

**LEMMA 3.1.** *Let  $e_1$  and  $e_2$  be two edges of  $T$  that cross each other. Then the endpoints of  $\delta(e_1, e_2)$  have different colors.*

**PROOF.** Let  $e_1 = (r_1, b_1)$  and  $e_2 = (r_2, b_2)$ . Suppose, for the sake of contradiction, that the endpoints of  $\delta(e_1, e_2)$  are of the same color, w.l.o.g. red. Then the endpoints of  $\delta(e_1, e_2)$  are  $r_1$  and  $r_2$  as in Figure 4. By definition,  $\delta(e_1, e_2)$  does not contain  $e_1$  and  $e_2$ . In this setting, we can replace edges  $(r_1, b_1)$  and  $(r_2, b_2)$  of  $T$  by two new edges  $(r_1, b_2)$  and  $(r_2, b_1)$  and obtain a new bichromatic spanning tree  $T'$ . By the triangle inequality (applied to each of the two triangles induced by the crossing), the total length of the two new edges is smaller than the total length of the two original edges. Hence  $T'$  is shorter than  $T$ , contradicting the minimality of  $T$ .  $\square$

**THEOREM 3.2.** *Every minimum bichromatic spanning tree is quasi-plane.*

**PROOF.** Let  $T$  be a Euclidean minimum bichromatic spanning tree. We prove that no three edges of  $T$  can pairwise cross each other. This will imply that  $T$  is quasi-plane. The proof proceeds by contradiction. Suppose that three edges of  $T$ , say  $e_1 = (r_1, b_1)$ ,  $e_2 = (r_2, b_2)$  and  $e_3 = (r_3, b_3)$ , pairwise cross each other as in Figure 5. We consider the following two cases:

1.  $\delta(e_i, e_j)$  contains  $e_k$  for some permutation of the indices with  $\{i, j, k\} = \{1, 2, 3\}$ .

After a suitable relabeling assume that  $\delta(e_1, e_2)$  contains  $e_3$ . Then  $\delta(e_1, e_3)$  and  $\delta(e_2, e_3)$  are sub-paths of  $\delta(e_1, e_2)$  and they do not contain  $e_2$  and  $e_1$  respectively. Assume without loss of generality (and by Lemma 3.1) that the endpoints of  $\delta(e_1, e_2)$  are  $r_2$  and  $b_1$  as in Figure 5(a);  $\delta(e_1, e_2)$  is highlighted in the figure. For the rest of our argument we will use a result of [20] which states that for an even number of (monochromatic) points in the plane, a perfect matching with pairwise crossing edges is the unique maximum-weight matching (without considering the colors). This means that  $M = \{e_1, e_2, e_3\}$  is the maximum matching for the point set  $\{r_1, r_2, r_3, b_1, b_2, b_3\}$ . Therefore  $M$  is longer than  $R = \{(r_1, b_3), (r_2, b_1), (r_3, b_2)\}$ , which is a

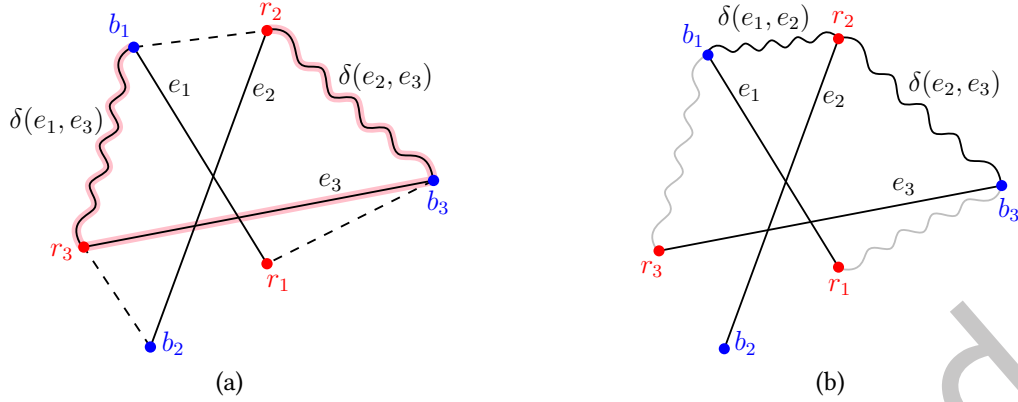


Fig. 5. (a) Replacing  $e_1, e_2, e_3$  by  $(r_1, b_3), (r_3, b_2), (r_2, b_1)$ ; the highlighted path is  $\delta(e_1, e_2)$ . (b) Getting a cycle in the union of  $\delta(e_1, e_2), \delta(e_2, e_3), \delta(e_1, e_3)$ , together with  $e_1$  or  $e_3$ . Gray paths represent the two possible choices for  $\delta(e_1, e_3)$ .

(bichromatic) matching for the same points. By replacing the edges of  $M$  in  $T$  with the edges of  $R$ , we obtain a shorter tree  $T'$ , contradicting the minimality of  $T$ . To verify that  $T'$  is a tree imagine replacing the edges one at a time. If we add  $(r_1, b_3)$  first, then we create a cycle that contains  $e_1$ , and thus by removing  $e_1$  we obtain a valid tree. Similarly we can replace  $e_2$  by  $(r_3, b_2)$  and  $e_3$  by  $(r_2, b_1)$ .

2.  $\delta(e_i, e_j)$  does not contain  $e_k$  for any permutation of the indices with  $\{i, j, k\} = \{1, 2, 3\}$ .

Consider the path  $\delta(e_1, e_2)$  and assume without loss of generality (and by Lemma 3.1) that its endpoints are  $r_2$  and  $b_1$  as in Figure 5(b). Now consider  $\delta(e_2, e_3)$ . This path cannot have  $r_3$  and  $b_2$  as its endpoints because otherwise the path  $\delta(e_1, e_3)$  would contain  $e_2$ , contradicting the assumption of the current case. Therefore the endpoints of  $\delta(e_2, e_3)$  are  $r_2$  and  $b_3$ . Now consider the path  $\delta(e_1, e_3)$ . We only have to consider  $\{r_1, b_3\}$  and  $\{b_1, r_3\}$  as endpoints for  $\delta(e_1, e_3)$  because of Lemma 3.1. If its endpoints are  $r_1$  and  $b_3$ , then the union of  $\delta(e_1, e_2), \delta(e_2, e_3)$ , and  $\delta(e_1, e_3)$  contains a path between  $r_1$  and  $b_1$  that does not go through  $e_1$ ; the union of this path and  $e_1$  is a cycle in  $T$ . Similarly, if the endpoints of  $\delta(e_1, e_3)$  are  $r_3$  and  $b_1$ , then the union of  $\delta(e_1, e_2), \delta(e_2, e_3), \delta(e_1, e_3)$ , and  $e_3$  contains a cycle. Both cases lead to a contradiction as  $T$  has no cycle.  $\square$

### 3.2 Maximum number of crossings

Given that a MinBST is quasi-plane (Theorem 3.2), one wonders how many crossings it can have. As illustrated in Figure 1, the number of crossings per edge can be linear in the number of points, and the total number of crossings can be quadratic. We give tight upper bounds for both quantities (Propositions 3.4–3.5), and also show that MinBST always has a crossing-free edge (Proposition 3.3).

**PROPOSITION 3.3.** *For every finite set of bichromatic points in the plane in general position, every MinBST contains a closest bichromatic pair as an edge. Moreover, no such edge is intersected by other edges of the MinBST.*

**PROOF.** We prove both parts by contradiction. For the first part assume that the MinBST does not contain an edge between any closest bichromatic pair. Let  $\{r_1, b_1\}$  be a closest bichromatic pair. By adding  $(r_1, b_1)$  to the tree we obtain a cycle in which  $(r_1, b_1)$  is the shortest edge. By removing any other edge from the cycle we obtain a bichromatic spanning tree of a shorter length. This contradicts the minimality of the original MinBST.

For the second part let  $\{r_1, b_1\}$  be a closest bichromatic pair that appears as an edge in the MinBST. Hence  $e_1 = (r_1, b_1)$  crosses some edge  $e_2 = (r_2, b_2)$ . Without loss of generality (and by Lemma 3.1), assume that the

endpoints of  $\delta(e_1, e_2)$  are  $r_1$  and  $b_2$ . If  $|r_2b_1| < |r_2b_2|$ , then by replacing  $(r_2, b_2)$  with  $(r_2, b_1)$  we obtain a shorter bichromatic spanning tree, a contradiction. Assume that  $|r_2b_1| \geq |r_2b_2|$ . Since  $\{r_1, b_1\}$  is a closest bichromatic pair we have  $|r_1b_2| \geq |r_1b_1|$ . Adding the two inequalities yields  $|r_1b_2| + |r_2b_1| \geq |r_1b_1| + |r_2b_2|$ . Note, however, that the vertices of any two crossing edges form a convex quadrilateral. By the triangle inequality, the total length of the two diagonals of a convex quadrilateral is strictly more than the total length of any pair of opposite edges, yielding  $|r_1b_2| + |r_2b_1| < |r_1b_1| + |r_2b_2|$ , a contradiction.  $\square$

**PROPOSITION 3.4.** *For every set of  $n \geq 2$  bichromatic points in the plane in general position, every MinBST has at most  $\lfloor n^2/4 \rfloor - n + 1$  crossings, and this bound is the best possible.*

**PROOF.** To verify that the claimed bound can be attained, consider the construction in Figure 1 where the two top clusters have  $\lfloor n/2 \rfloor - 1$  and  $\lceil n/2 \rceil - 1$  points. Then the total number of crossings in  $\text{MinBST}(S)$  is  $(\lfloor n/2 \rfloor - 1) \cdot (\lceil n/2 \rceil - 1)$ , which is equal to  $\lfloor n^2/4 \rfloor - n + 1$  because  $n$  is an integer.

For an upper bound, let  $S$  be a set of  $n$  bichromatic points in general position. Define the *crossing graph*  $G_{\text{cr}}$  of  $\text{MinBST}(S)$ , where the vertices of  $G_{\text{cr}}$  correspond to the edges of  $\text{MinBST}(S)$  and edges of  $G_{\text{cr}}$  represent crossings between the edges of  $\text{MinBST}(S)$ . Note that  $G_{\text{cr}}$  has  $n - 1$  vertices where one of them is of degree 0 by Proposition 3.3. By Theorem 3.2,  $G_{\text{cr}}$  is triangle-free. Therefore, by Turán's theorem [7],  $G_{\text{cr}}$  has at most  $(\lfloor n/2 \rfloor - 1) \cdot (\lceil n/2 \rceil - 1)$  edges. Consequently,  $\text{MinBST}(S)$  has at most this many crossings.  $\square$

**PROPOSITION 3.5.** *For every set of  $n \geq 3$  bichromatic points in the plane in general position, every edge of a MinBST crosses at most  $n - 3$  other edges, and this bound is the best possible.*

**PROOF.** To verify that the claimed bound can be attained, consider the construction in Figure 1 which contains a cluster with 3 red points and a cluster with 3 blue points. Replace the red cluster by a single point and the blue cluster by  $n - 3$  points. Then all points in the blue cluster have degree 1 in any  $\text{MinBST}(S)$ ; these  $n - 3$  leaves all cross one edge of the  $\text{MinBST}(S)$ .

For the upper bound, notice that a MinBST has  $n - 1$  edges, one of which is crossing-free by Proposition 3.3. Consequently, an edge in a MinBST can cross at most  $n - 3$  other edges.  $\square$

## 4 CONCLUSIONS AND OPEN PROBLEMS

We conclude with a collection of open problems raised by our results. We have presented a  $O(\log n)$ -approximation algorithm for the MinPBST problem for a set of  $n$  bichromatic points in the plane, and showed that  $\rho_n \leq O(\log n)$ . Recall that the current best lower bound is  $\rho_n \geq 3/2$  for all  $n \geq 4$  [29]. It remains open whether a constant-factor approximation is possible, whether the problem is APX-hard, and whether  $\rho_n$  is bounded by a constant.

There is an  $O(n^2)$  gap between the running time of our randomized algorithm and the deterministic version. It would be interesting to narrow this gap.

It is also natural to investigate whether there is an (approximation) algorithm that, given a bichromatic point set and an integer  $d$ , finds a *minimum plane bichromatic tree of maximum degree at most  $d$*  (or reports that none exists). It is known that any set of  $n$  red and  $n$  blue points in general position admits a plane bichromatic spanning tree of maximum degree at most three [34]; but there are  $n$  red and  $n$  blue points in convex position that do not admit a bichromatic plane spanning path [8]. For the general case of  $n$  red and  $m$  blue points, with  $n \geq m$ , there exists a plane bichromatic spanning tree of maximum degree at most  $\max\{3, \lceil n-1/m \rceil + 1\}$  and this is the best upper bound [19].

We have shown that MinBST is quasi-plane, which means that the crossing graph  $G_{\text{cr}}$  of MinBST is triangle-free. Figure 1 shows that  $G_{\text{cr}}$  can have 4-cycles (and even cycles of any lengths). Can the crossing graph  $G_{\text{cr}}$  of MinBST contain an odd cycle (e.g., a 5-cycle)? Can every MinBST be decomposed into a constant number of planar straight-line graphs?

## ACKNOWLEDGEMENT

This work was initiated at the Eleventh Annual Workshop on Geometry and Graphs, held at the Bellairs Research Institute in Holetown, Barbados in March, 2024. The authors thank the organizers and the participants. H. A. Akitaya has been partially supported by the NSF award CCF-2348067. A. Biniaz was supported in part by NSERC. Research by Cs. D. Tóth was supported in part by the NSF award DMS-2154347.

## REFERENCES

- [1] Manuel Abellanas, Jesus García-Lopez, Gregorio Hernández-Peñalver, Marc Noy, and Pedro A. Ramos. Bipartite embeddings of trees in the plane. *Discret. Appl. Math.*, 93(2-3):141–148, 1999. doi:10.1016/S0166-218X(99)00042-6.
- [2] A. Karim Abu-Affash, Sujoy Bhore, Paz Carmi, and Joseph S. B. Mitchell. Planar bichromatic bottleneck spanning trees. *J. Comput. Geom.*, 12(1):109–127, 2021. doi:10.20382/JOCG.V12I1A5.
- [3] Eyal Ackerman. On the maximum number of edges in topological graphs with no four pairwise crossing edges. *Discret. Comput. Geom.*, 41(3):365–375, 2009. doi:10.1007/S00454-009-9143-9.
- [4] Pankaj K. Agarwal. Partitioning arrangements of lines I: An efficient deterministic algorithm. *Discret. Comput. Geom.*, 5:449–483, 1990. doi:10.1007/BF02187805.
- [5] Pankaj K. Agarwal, Boris Aronov, János Pach, Richard Pollack, and Micha Sharir. Quasi-planar graphs have a linear number of edges. *Combinatorica*, 17(1):1–9, 1997. doi:10.1007/BF01196127.
- [6] Oswin Aichholzer, Johannes Obenaus, Joachim Orthaber, Rosna Paul, Patrick Schnider, Raphael Steiner, Tim Taubner, and Birgit Vogtenhuber. Edge partitions of complete geometric graphs. In *Proc. 38th Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPIcs*, pages 6:1–6:16. Schloss Dagstuhl, 2022. doi:10.4230/LIPICS.SOCG.2022.6.
- [7] Martin Aigner and Günter M. Ziegler. Turán’s graph theorem. In *Proofs from THE BOOK*, chapter 41, pages 285–289. 6th edition, 2018. doi:10.1007/978-3-662-57265-8\_41.
- [8] Jin Akiyama and Jorge Urrutia. Simple alternating path problem. *Discret. Math.*, 84(1):101–103, 1990. doi:10.1016/0012-365X(90)90276-N.
- [9] Carlos Alegría, David Orden, Carlos Seara, and Jorge Urrutia. Separating bichromatic point sets in the plane by restricted orientation convex hulls. *J. Glob. Optim.*, 85(4):1003–1036, 2023. doi:10.1007/S10898-022-01238-9.
- [10] Patrizio Angelini, Michael A. Bekos, Franz J. Brandenburg, Giordano Da Lozzo, Giuseppe Di Battista, Walter Didimo, Michael Hoffmann, Giuseppe Liotta, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Simple  $k$ -planar graphs are simple  $(k + 1)$ -quasiplanar. *J. Comb. Theory B*, 142:1–35, 2020. doi:10.1016/J.JCTB.2019.08.006.
- [11] Bogdan Armaselu and Ovidiu Daescu. Dynamic minimum bichromatic separating circle. *Theor. Comput. Sci.*, 774:133–142, 2019. doi:10.1016/J.TCS.2016.11.036.
- [12] Boris Aronov, Paul Erdős, Wayne Goddard, Daniel J. Kleitman, Michael Klugerman, János Pach, and Leonard J. Schulman. Crossing families. *Combinatorica*, 14(2):127–134, 1994. doi:10.1007/BF01215345.
- [13] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.
- [14] Sanjeev Arora and Kevin L. Chang. Approximation schemes for degree-restricted MST and red-blue separation problems. *Algorithmica*, 40(3):189–210, 2004. doi:10.1007/s00453-004-1103-4.
- [15] Sayan Bandyapadhyay, Aritra Banik, Sujoy Bhore, and Martin Nöllenburg. Geometric planar networks on bichromatic collinear points. *Theor. Comput. Sci.*, 895:124–136, 2021. doi:10.1016/J.TCS.2021.09.035.
- [16] Sergei Bespamyatnikh, David G. Kirkpatrick, and Jack Snoeyink. Generalizing ham sandwich cuts to equitable subdivisions. *Discret. Comput. Geom.*, 24(4):605–622, 2000. doi:10.1007/s004540010065.
- [17] Ahmad Biniaz, Prosenjit Bose, Kimberly Crosbie, Jean-Lou De Carufel, David Eppstein, Anil Maheshwari, and Michiel H. M. Smid. Maximum plane trees in multipartite geometric graphs. *Algorithmica*, 81(4):1512–1534, 2019. doi:10.1007/S00453-018-0482-X.
- [18] Ahmad Biniaz, Prosenjit Bose, David Eppstein, Anil Maheshwari, Pat Morin, and Michiel H. M. Smid. Spanning trees in multipartite geometric graphs. *Algorithmica*, 80(11):3177–3191, 2018. doi:10.1007/S00453-017-0375-4.
- [19] Ahmad Biniaz, Prosenjit Bose, Anil Maheshwari, and Michiel H. M. Smid. Plane bichromatic trees of low degree. *Discret. Comput. Geom.*, 59(4):864–885, 2018. doi:10.1007/S00454-017-9881-Z.
- [20] Ahmad Biniaz, Anil Maheshwari, and Michiel Smid. Euclidean maximum matchings in the plane—local to global. In *Proceedings of the 17th International Symposium on Algorithms and Data Structures (WADS)*, volume 12808 of *LNCs*, pages 186–199. Springer, 2021. doi:10.1007/978-3-030-83508-8\_14.
- [21] Magdalene G. Borgelt, Marc J. van Kreveld, Maarten Löffler, Jun Luo, Damian Merrick, Rodrigo I. Silveira, and Mostafa Vahedi. Planar bichromatic minimum spanning trees. *J. Discrete Algorithms*, 7(4):469–478, 2009. doi:10.1016/J.JDA.2008.08.001.
- [22] Otakar Borůvka. O jistém problému minimálním. *Praca Moravske Prirodovedcke Spolecnosti*, 3(3):37–58, 1926.

- [23] Vasilis Capioleas and János Pach. A Turán-type theorem on chords of a convex polygon. *J. Comb. Theory B*, 56(1):9–15, 1992. doi:10.1016/0095-8956(92)90003-G.
- [24] Timothy M. Chan. On the bichromatic  $k$ -set problem. *ACM Trans. Algorithms*, 6(4):62:1–62:20, 2010. doi:10.1145/1824777.1824782.
- [25] Timothy M. Chan and Bryan T. Wilkinson. Bichromatic line segment intersection counting in  $O(n\sqrt{\log n})$  time. In *Proceedings of the 23rd Annual Canadian Conference on Computational Geometry (CCCG)*, Toronto, ON, 2011. URL: <https://cccg.ca/proceedings/2011/papers/paper83.pdf>.
- [26] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. Quadrees. In *Computational Geometry: Algorithms and Applications*, chapter 14, pages 307–322. Springer, Berlin, 2008. doi:10.1007/978-3-540-77974-2\_14.
- [27] Erik D. Demaine, Jeff Erickson, Ferran Hurtado, John Iacono, Stefan Langerman, Henk Meijer, Mark H. Overmars, and Sue Whitesides. Separating point sets in polygonal environments. *Int. J. Comp. Geom. Appl.*, 15(4):403–420, 2005. doi:10.1142/S0218195905001762.
- [28] Jacob Fox, János Pach, and Andrew Suk. Quasiplanar graphs, string graphs, and the Erdős-Gallai problem. In *Proceedings of the 30th International Symposium on Graph Drawing and Network Visualization (GD)*, pages 219–231, 2022. doi:10.1007/978-3-031-22203-0\_16.
- [29] Magdalene Grantson, Henk Meijer, and David Rappaport. Bi-chromatic minimum spanning trees. In *Proceedings of the 21st European Workshop on Computational Geometry (EwCG)*, pages 199–202, Eindhoven, 2005. URL: <https://www.win.tue.nl/EwCG2005/Proceedings/51.pdf>.
- [30] John Hershberger and Subhash Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32(2):249–267, 1992. doi:10.1007/BF01994880.
- [31] Michael Hoffmann, Bettina Speckmann, and Csaba D. Tóth. Pointed binary encompassing trees: Simple and optimal. *Comput. Geom.*, 43(1):35–41, 2010. doi:10.1016/j.comgeo.2006.12.005.
- [32] Michael Hoffmann and Csaba D. Tóth. Vertex-colored encompassing graphs. *Graphs and Combinatorics*, 30(4):933–947, 2014. doi:10.1007/s00373-013-1320-1.
- [33] Ferran Hurtado, Mikio Kano, David Rappaport, and Csaba D. Tóth. Encompassing colored planar straight line graphs. *Comput. Geom.*, 39(1):14–23, 2008. doi:10.1016/J.COMGEO.2007.05.006.
- [34] Atsushi Kaneko. On the maximum degree of bipartite embeddings of trees in the plane. In *Discrete and Computational Geometry (JCDCG)*, volume 1763 of *LNCS*, pages 166–171, Heidelberg, 1998. Springer. doi:10.1007/978-3-540-46515-7\_13.
- [35] Atsushi Kaneko and Mikio Kano. Discrete geometry on red and blue points in the plane — a survey. In Boris Aronov, Saugata Basu, János Pach, and Micha Sharir, editors, *Discrete and Computational Geometry*, volume 25 of *Algorithms and Combinatorics*, pages 551–570. Springer Berlin Heidelberg, 2003. doi:10.1007/978-3-642-55566-4\_25.
- [36] Mikio Kano, Kazuhiro Suzuki, and Miyuki Uno. Properly colored geometric matchings and 3-trees without crossings on multicolored points in the plane. In *Discrete and Computational Geometry and Graphs (JCDCG)*, *LNCS*, pages 96–111, Cham, 2013. Springer. doi:10.1007/978-3-319-13287-7\_9.
- [37] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. doi:10.1090/S0002-9939-1956-0078686-7.
- [38] Harry G. Mairson and Jorge Stolfi. Reporting and counting intersections between two sets of line segments. In Rae A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, volume 40 of *NATO ASI Series*, pages 307–325. Springer, Heidelberg, 1988. doi:10.1007/978-3-642-83539-1\_11.
- [39] János Pach, Farhad Shahrokhi, and Mario Szegedy. Applications of the crossing number. *Algorithmica*, 16(1):111–117, 1996. doi:10.1007/BF02086610.
- [40] János Pach, Natan Rubin, and Gábor Tardos. Planar point sets determine many pairwise crossing segments. *Advances in Mathematics*, 386:107779, 2021. doi:10.1016/j.aim.2021.107779.
- [41] Robert C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36, 1957.

Received 22 January 2025; revised 22 January 2025; accepted 2 July 2025