RESEARCH-ARTICLE

Long Plane Trees

SERGIO CABELLO, University of Ljubljana, Ljubljana, Slovenia

MICHAEL H W HOFFMANN, Swiss Federal Institute of Technology, Zurich, Zurich, ZH, Switzerland

KATHARINA KLOST, Free University of Berlin, Berlin, Berlin, Germany

WOLFGANG JOHANN HEINRICH MULZER, Free University of Berlin, Berlin, Berlin, Germany

JOSEF TKADLEC, Charles University, Prague, Czech Republic

Published: 07 October 2025
Online AM: 04 September 2025
Accepted: 31 July 2025
Revised: 21 May 2025
Received: 30 April 2024

Citation in BibTeX format

ACM Transactions on Algorithms, Volume 22, Issue 1 (January 2026)
https://doi.org/10.1145/3765740
EISSN: 1549-6333

Long Plane Trees

SERGIO CABELLO, Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia and Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia
MICHAEL HOFFMANN, ETH Zürich, Zürich, Switzerland
KATHARINA KLOST and WOLFGANG MULZER, Institut für Informatik, Freie Universität Berlin, Berlin, Germany
JOSEF TKADLEC, Computer Science Institute, Charles University, Praha, Czech Republic

In the *longest plane spanning tree* problem, we are given a finite planar point set $\mathcal{P}$, and our task is to find a plane (i.e., noncrossing) spanning tree for $\mathcal{P}$ with maximum total Euclidean edge length. Despite more than two decades of research, it remains open whether this problem is NP-hard. Thus, previous results have focused on polynomial-time algorithms that produce plane trees whose total edge length approximates OPT, the maximum possible length. The approximate trees in these algorithms all have small unweighted diameter, typically two to four. It is natural to ask whether this is a common feature of longest plane spanning trees, or an artifact of the specific approximation algorithms.

We provide three results to elucidate the interplay between the approximation guarantee and the unweighted diameter of the approximate trees. First, we describe a polynomial-time algorithm to construct a plane tree with diameter at most four and total edge length at least $0.546 \cdot \mathrm{OPT}$. This constitutes a substantial improvement over the state of the art. Second, we show that a longest plane tree among those with diameter at most three can be found in polynomial time. Third, for any candidate diameter $d \geq 3$, we provide upper bounds on the approximation factor that can be achieved by a longest plane tree with diameter at most d (compared to a longest plane tree without constraints).

CCS Concepts: • **Theory of computation → Routing and network design problems**; **Computational geometry**; • **Mathematics of computing → Trees**;

Additional Key Words and Phrases: geometric network design, spanning trees, plane straight-line graphs, approximation algorithms

---

A preliminary version appeared as S. Cabello, M. Hoffmann, K. Klost, W. Mulzer, and J. Tkadlec. *Long Plane Trees* in *Proc. 38th SoCG*, pp. 23:1–23:17.

S. Cabello was supported in part by the Slovenian Research Agency (P1-0297, J1-9109, J1-8130, J1-8155, J1-1693, J1-2452, N1-0218, N1-0285) and in part by the European Union (ERC, KARST, project number 101071836). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. M. Hoffmann was supported by the Swiss National Science Foundation within the collaborative DACH project Arrangements and Drawings as SNSF Project 200021E-171681. W. Mulzer was supported in part by ERC StG 757609. J. Tkadlec was supported by Charles Univ. projects UNCE 24/SCI/008 and PRIMUS 24/SCI/012.

Authors' Contact Information: Sergio Cabello, Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia and Institute of Mathematics Physics and Mechanics, Ljubljana, Slovenia; e-mail: sergio.cabello@fmf.uni-lj.si; Michael Hoffmann, ETH Zürich, Zürich, Switzerland; e-mail: hoffmann@inf.ethz.ch; Katharina Klost, Institut für Informatik, Freie Universität Berlin, Berlin, Germany; e-mail: katharina.klost@fu-berlin.de; Wolfgang Mulzer (corresponding author), Institut für Informatik, Freie Universität Berlin, Berlin, Germany; e-mail: mulzer@inf.fu-berlin.de; Josef Tkadlec, Computer Science Institute, Charles University, Praha, Czech Republic; e-mail: josef.tkadlec@iuuk.mff.cuni.cz.

ACM 1549-6333/2025/10-ART5
https://doi.org/10.1145/3765740

ACM Reference format:
Sergio Cabello, Michael Hoffmann, Katharina Klost, Wolfgang Mulzer, and Josef Tkadlec. 2025. Long Plane Trees. *ACM Trans. Algor.* 22, 1, Article 5 (October 2025), 40 pages.
https://doi.org/10.1145/3765740

---

1 Introduction

Geometric network design is a common and well-studied task in computational geometry and combinatorial optimization [19, 22, 26, 27]. In this family of problems, we are given a set $\mathcal{P}$ of points in general position, and our task is to connect $\mathcal{P}$ into a (geometric) graph that has certain favorable properties. Not surprisingly, this general question has captivated the attention of researchers for a long time, and we can find countless variants, depending on which restrictions we put on the graph that connects $\mathcal{P}$ and which criteria of this graph we would like to optimize. Typical graph classes of interest include matchings, paths, cycles, trees, or general *plane* (*noncrossing*) graphs, i.e., graphs, whose straight-line embedding on $\mathcal{P}$ does not contain any edge crossings. Typical quality criteria include the total edge length [3, 16, 25, 30], the maximum length (bottleneck) edge [6, 18], the maximum degree [4, 12, 20, 33], the dilation [19, 28, 31], or the stabbing number [29, 36] of the graph.

Many famous problems from computational geometry fall into this general setting. For example, if our goal is to minimize the total edge length, while restricting ourselves to paths, trees, or triangulations, respectively, we are faced with the venerable problems of finding an optimum TSP tour [22], a Euclidean minimum spanning tree [16], or a minimum weight triangulation [30] for $\mathcal{P}$. These three examples also illustrate the wide variety of complexity aspects that we may encounter in geometric design problems: the Euclidean TSP problem is known to be NP-hard [32], but it admits a PTAS [3, 25]. On the other hand, it is possible to find a Euclidean minimum spanning tree for $\mathcal{P}$ in polynomial time [16] (even though, curiously, the associated decision problem is not known to be solvable by a polynomial-time Turing machine, see, e.g., [9]). The minimum weight triangulation problem is also known to be NP-hard [30], but the existence of a PTAS is still open; however, a QPTAS is known [35].

In this work, we are interested in the interaction of two specific requirements for a geometric design problem, namely the two objectives of obtaining a plane graph and of optimizing the total edge length. For the case that we want to *minimize* the total edge length of the resulting graph, these two goals are often in perfect harmony: the shortest Euclidean TSP tour and the shortest Euclidean minimum spanning tree are automatically plane, as can be seen by a simple application of the triangle inequality. In contrast, if our goal is to *maximize* the total edge length, while obtaining a plane graph, much less is known.

This family of problems was studied by Alon et al. [1], who considered the problems of computing a longest plane matching, a longest plane Hamiltonian path, and a longest plane spanning tree for a planar point set $\mathcal{P}$ in general position. They conjectured that these three problems are all NP-hard, but as far as we know, this is still open. The situation is similar for the problem of finding a *maximum weight triangulation* for $\mathcal{P}$: here, we have neither an NP-hardness proof, nor a polynomial time algorithm [13]. If we omit the planarity condition, then the problem of finding a longest Hamiltonian path (the *geometric maximum TSP problem*) is known to be NP-hard in dimension three and above, while the two-dimensional case remains open [5]. On the other hand, we can find a longest (typically not plane) tree on $\mathcal{P}$ in polynomial time, using classic greedy algorithms [15], and a similar result holds for finding a longest (typically not plane) matching on $\mathcal{P}$.

Longest Plane Spanning Trees. We focus on the specific problem of finding a longest plane (i.e., noncrossing) tree for a given set $\mathcal{P}$ of $n \geq 3$ points in the plane in general position (that is, no three points in $\mathcal{P}$ are collinear). Such a tree is necessarily spanning. The general position assumption was also used in previous work on this problem [1, 17]. Without it, one should specify whether overlapping edges are allowed. This is an additional complication that we would like to avoid.

If $\mathcal{P}$ is in convex position, then the longest plane tree for $\mathcal{P}$ can be found in polynomial time on a real RAM, by adapting standard dynamic programming methods for plane structures on point sets in convex position [21, 23]. On the other hand, for an arbitrary point set $\mathcal{P}$, the problem is conjectured—but not known—to be NP-hard [1]. Hence, past results have focused on designing polynomial-time approximation algorithms. Typically, these algorithms construct several "simple" spanning trees for $\mathcal{P}$ of small (unweighted) diameter, and one then argues that at least one such tree is sufficiently long. In a seminal work, Alon et al. [1] showed that a longest star (a plane tree with diameter two) on $\mathcal{P}$ yields a 0.5-approximation for the longest (not necessarily plane) spanning tree of $\mathcal{P}$. They further argued that this bound is essentially tight for point sets that consist of two large clusters far away from each other. Dumitrescu and Tóth [17] refined this algorithm by adding two additional families of candidate trees, now with diameter four. They showed that at least one member of this extended range of candidates constitutes a 0.502-approximation, which was further improved to 0.503 by Biniaz et al. [8]. In all these results, the approximation factor is analyzed by comparing to the length of a longest (typically not plane) spanning tree. Such a tree may be longer by a factor of up to $\pi/2$, $\pi/2 > 1.5$ than a maximum-length plane tree [1], as witnessed by, e.g., a large set of points spaced uniformly on a unit circle. While the ratio between the lengths of the longest plane tree and the longest (possibly crossing) tree is an interesting quantity in itself, the original objective is to construct a longest plane tree, and thus, it is better to compare the length of the constructed plane trees against the true optimum, that is, against the length of a longest plane tree. Considering certain trees of diameter at most five, a superset of the authors of this paper managed to compare against the longest plane tree and pushed the approximation factor to 0.512 [11]. This was subsequently improved even further to 0.519 by Biniaz [7].

Our Results. We provide a deeper study of the interplay between the approximation factor and the diameter of the candidate trees. First, we give a polynomial-time algorithm to find a tree of diameter at most four that guarantees an approximation factor of roughly 0.546, a substantial improvement over the previous bounds.

Theorem 1.1. *For any finite point set $\mathcal{P}$ in general position (the set $\mathcal{P}$ contains no three collinear points), we can compute in polynomial time a plane tree of Euclidean length at least $f \cdot \mathrm{OPT}$, where OPT denotes the length of a longest plane tree on $\mathcal{P}$ and $f > 0.5467$ is the fourth smallest real root of the polynomial*

$$P(x) = -80 + 128x + 504x^2 - 768x^3 - 845x^4 + 1096x^5 + 256x^6.$$

The algorithm of Theorem 1.1 constructs n^2 plane trees: n stars and $n(n-1)$ general trees of diameter at most four. We show that one of these trees is always sufficiently long. The algorithm is very simple, but its analysis uses several geometric insights. The polynomial $P(x)$ comes from optimizing several constants that appear in the algorithm.

Second, we characterize longest plane trees for point sets in convex position. A *caterpillar* is a tree T that contains a path S, the *spine*, so that every vertex of $T \setminus S$ is adjacent to a vertex in S. A tree T that is straight-line embedded on a point set $\mathcal{P}$ in convex position is a *zigzagging caterpillar* if its edges split the convex hull of $\mathcal{P}$ into faces that are all triangles. Our next two theorems show that the longest plane trees for point sets in convex position are given exactly by caterpillar graphs.

Theorem 1.2. *Let $\mathcal{P}$ be a finite point set in convex position in the plane. Then, every longest plane tree on $\mathcal{P}$ is a zigzagging caterpillar.*

Theorem 1.3. *For any caterpillar C, there exists a point set $\mathcal{P}$ in convex position such that the longest tree for $\mathcal{P}$ is unique and isomorphic to C.*

In particular, Theorem 1.3 implies that the diameter of a (unique) longest plane tree can be arbitrarily large. As a consequence, we obtain an upper bound on the approximation factor $\mathrm{BoundDiam}(d)$ that can be achieved by a plane tree of diameter at most d.

Theorem 1.4. *For every $d \geq 2$, there exists a point set $\mathcal{P}$ in convex position so that every plane tree of diameter at most d on $\mathcal{P}$ is at most*

$$1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3)$$

times as long as the length $|T_{\mathrm{OPT}}|$ of a longest (general) plane tree on $\mathcal{P}$. It follows that

$$\mathrm{BoundDiam}(d) \leq 1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3).$$

We have better bounds for small values of d. For $d = 2$, it is easy to see that $\mathrm{BoundDiam}(2) \leq 1/2$: put two groups of roughly half of the points sufficiently far from each other [1]. For $d = 3$, we can show $\mathrm{BoundDiam}(3) \leq 5/6$.

Theorem 1.5. *For every $\varepsilon > 0$, there exists a point set $\mathcal{P}$ in convex position such that every longest plane tree of diameter three on $\mathcal{P}$ is at most $(5/6) + \varepsilon$ times as long as every longest (general) plane tree on $\mathcal{P}$.*

Third, we give polynomial-time algorithms for finding a longest plane tree among those of diameter at most three and among a special class of trees of diameter at most four. Note that in contrast to diameter two, the number of spanning trees of diameter at most three is exponential in the number of points.

Theorem 1.6. *For any set $\mathcal{P}$ of n points in general position, a longest plane tree of diameter at most three on $\mathcal{P}$ can be computed in $O(n^4)$ time.*

Theorem 1.7. *For any set $\mathcal{P}$ of points in general position and for any three specified points on the boundary of the convex hull of $\mathcal{P}$, we can compute in polynomial time the longest plane tree such that each edge is incident to at least one of the three specified points.*

The algorithms are based on dynamic programming. Even though the length OPT_3 of a longest plane tree of diameter at most three can be computed in polynomial time, we do not know the corresponding approximation factor $\mathrm{BoundDiam}(3)$ with respect to OPT. The best bounds we are aware of are $1/2 \leq \mathrm{BoundDiam}(3) \leq 5/6$. The lower bound follows from [1], the upper bound is from Theorem 1.5. We conjecture that OPT_3 actually gives a better approximation factor than the tree constructed in Theorem 1.1—but we are unable to prove this.

Finally, a natural way to design an algorithm for the longest plane spanning tree problem is the following local search heuristic [37]: start with an arbitrary plane tree T, and while it is possible, apply the following *local improvement rule*: if there are two edges e, f on $\mathcal{P}$ such that $(T \setminus \{e\}) \cup \{f\}$ is a plane spanning tree for $\mathcal{P}$ that is longer than T, replace e by f. Once no further local improvements are possible, output the current tree T. We show that for some point sets, this algorithm fails to compute the optimum answer as it may "get stuck" in a local optimum (see

Fig. 1. A tree S_a and a tree $T_{a,b}$.

Lemma 5.1 in Section 5). This holds regardless of how the edges that are swapped are chosen. This suggests that a natural local search approach does not yield an algorithm for the problem.

Preliminaries and Notation. Let $\mathcal{P} \subset \mathbb{R}^2$ be a set of n points in the plane, so that no three points in $\mathcal{P}$ are collinear. For any spanning tree T on $\mathcal{P}$, we denote by $|T|$ the total Euclidean edge length of T. Let OPT be the maximum Euclidean length of a plane (i.e., noncrossing) spanning tree on $\mathcal{P}$, and $\mathrm{OPT}_{\mathrm{cr}}$ the maximum length of a (possibly crossing) spanning tree on $\mathcal{P}$.

As in previous algorithms [1, 7, 8, 11, 17], we make extensive use of stars. For any point $p \in \mathcal{P}$, the *star* S_p rooted at p is the tree that connects p to all other points of $\mathcal{P}$.

We also need a notion of "flat" point sets. A point set $\mathcal{P}$ is *flat* if $\mathrm{diam}(\mathcal{P}) \geq 1$ and all y-coordinates in $\mathcal{P}$ are essentially negligible, that is, their absolute values are bounded by an infinitesimal $\varepsilon > 0$. In a flat point set, we can approximate the length of an edge by the difference between the x-coordinates of its endpoints: the error becomes arbitrarily small as $\varepsilon \longrightarrow 0$.

Lastly, $D(p, r)$ denotes a closed disk with center p and radius r, while $\partial D(p, r)$ is its boundary: a circle of radius r centered at p.

2 A General Approximation Algorithm

We present a polynomial-time algorithm that yields an f-approximation, for $f \doteq 0.5467$, of a longest plane tree for general point sets and a $(2/3)$-approximation for flat point sets.

Let $\mathcal{P}$ be the input point set (in general position). Our algorithm considers two kinds of plane spanning trees for $\mathcal{P}$. First, for every point $a \in \mathcal{P}$, let S_a be the star rooted at a, as defined in Section 1. Second, for any distinct points $a, b \in \mathcal{P}$, we define a tree $T_{a,b}$ as follows (see Figure 1): let $\mathcal{P}_a$ be the points of $\mathcal{P}$ that are closer to a than to b, and let $\mathcal{P}_b = \mathcal{P} \setminus \mathcal{P}_a$. First, we connect a to every point in $\mathcal{P}_b$. Then, we connect each point of $\mathcal{P}_a \setminus \{a\}$ to some point of $\mathcal{P}_b$, according to the following rule: the rays $\overrightarrow{av}$ for $v \in \mathcal{P}_b$ together with the opposing ray of $\overrightarrow{ab}$ partition the plane into convex wedges with common apex a. For each such wedge W, let $b_W \in \mathcal{P}_b$ the point on a bounding ray of W that forms the smaller (convex) angle with $\overrightarrow{ab}$. Within each wedge W, we connect all points of $W \cap (\mathcal{P}_a \setminus \{a\})$ to b_W. The resulting tree $T_{a,b}$ has diameter at most four, and it is plane, because the interiors of the wedges are pairwise disjoint, and we add a star within each wedge.

In general, the trees $T_{a,b}$ and $T_{b,a}$ are different. However, for $\mathcal{P}_a = \{a\}$, both $T_{a,b}$ and $T_{b,a}$ coincide with the star S_a.

Now, our algorithm `AlgSimple(𝒫)` constructs the n stars S_a, for $a \in \mathcal{P}$, and the $n(n-1)$ trees $T_{a,b}$, for distinct $a, b \in \mathcal{P}$, and it computes the total edge length for each of them. It returns the longest of these n^2 trees as the desired approximation. We denote this resulting tree as T_{ALG}.

The algorithm `AlgSimple(𝒫)` runs in polynomial time on a real RAM, as there are n^2 relevant trees, each of which can be constructed in polynomial time.[1]

---

[1] The *real RAM* is the standard computational model in computational geometry [34]. It allows constant-time operations on real numbers. For our problem, this is relevant, because in order to find the longest tree, we must compare sums of $\Theta(n)$ distances, and these distances are real numbers derived from the input. We do not know how to do this in polynomial time on a Turing machine (where the inputs are rational numbers) [10]. However, the sum of distances can be *approximated*

It remains to analyze the approximation guarantee of `AlgSimple`. The main result of this section (Theorem 1.1) states that for any point set $\mathcal{P}$, we have $|T_{\text{ALG}}| > 0.5467 \cdot \text{OPT}$. The proof is rather involved. As a warm-up for the full proof, we first show a stronger result for the special case of *flat* point sets (cf. Section 1): if $\mathcal{P}$ is flat, we have $|T_{\text{ALG}}| \geq (2/3) \cdot \text{OPT}_{\text{cr}}$, where OPT_{cr} is the length of a longest (possibly crossing) spanning tree.

Theorem 2.1. *Suppose $\mathcal{P}$ is flat. Then $|T_{\text{ALG}}| \geq \frac{2}{3} \cdot \text{OPT}_{\text{cr}} \geq \frac{2}{3} \cdot \text{OPT}$.*

Proof. Since $\text{OPT}_{\text{cr}} \geq \text{OPT}$, it suffices to prove the first inequality. Recall that in a flat point set, the y-coordinates of the points are essentially negligible. Thus, let a be the leftmost point and b be the rightmost point of $\mathcal{P}$, and let $m = (a+b)/2$ be the midpoint between a and b. Let T_{cr} be the tree on $\mathcal{P}$ where all points to the right of m are connected to a, and all points to the left of m are connected to b.[2] Then as the greedy algorithm finds an optimal tree, T_{cr} is a tree that achieves OPT_{cr}, In particular, for any edge pq that is not in the T_{cr}, the *fundamental* cycle that pq makes with T_{cr} has pq as a shortest edge. Now, to analyze the quality of the approximation, we consider the four trees S_a, $T_{a,b}$, $T_{b,a}$, and S_b. Our goal is to show that

$$\max\{|S_a|, |T_{a,b}|, |T_{b,a}|, |S_b|\} \geq \frac{2}{3} \cdot \text{OPT}_{\text{cr}}, \tag{1}$$

which implies the desired result. For this, we define a probability distribution $\mathcal{D}$ on $\{S_a, T_{a,b}, T_{b,a}, S_b\}$, and we show that

$$\mathbf{E}_{T\sim\mathcal{D}}[|T|] \geq \frac{2}{3} \cdot \text{OPT}_{\text{cr}}, \tag{2}$$

which immediately implies (1). For (2), we use the distribution $\mathcal{D}$ with $\Pr_{\mathcal{D}}[S_a] = \Pr_{\mathcal{D}}[S_b] = 1/6$ and $\Pr_{\mathcal{D}}[T_{a,b}] = \Pr_{\mathcal{D}}[T_{b,a}] = 1/3$. Our strategy is to consider the edges individually and to use linearity of expectation to achieve the overall bound (2). In particular, for $p \in \mathcal{P} \setminus \{a\}$ and $T \in \{S_a, T_{a,b}, T_{b,a}, S_b, T_{\text{cr}}\}$, let $\ell_T(p)$ be the length of the first edge on the path from p to a in T. By the definition of T_{cr}, we have

$$\ell_{T_{\text{cr}}}(p) = \max\{\|pa\|, \|pb\|\}, \tag{3}$$

for all $p \in \mathcal{P} \setminus \{a\}$. Next, we analyze $\mathbf{E}_{T\sim\mathcal{D}}[\ell_T(p)]$, for $p \in \mathcal{P} \setminus \{a\}$. Suppose first that p lies to the right of m. Since S_a and S_b are stars, we have $\ell_{S_a}(p) = \|pa\|$ and $\ell_{S_b}(p) = \|pb\|$. Furthermore, by our assumption on p, we have $\ell_{T_{a,b}}(p) = \|pa\|$, and $\ell_{T_{b,a}}(p) \geq \|mp\|$. Now, since m is the midpoint between a and b, we have $\|am\| = \|mb\|$, and hence

$$\|pa\| = \|am\| + \|mp\| = \|mb\| + \|mp\| = \|mp\| + \|pb\| + \|mp\| = 2\|mp\| + \|pb\|. \tag{4}$$

Altogether, this gives

$$\begin{aligned}
\mathbf{E}_{T\sim\mathcal{D}}[\ell_T(p)] &= \frac{1}{6} \cdot \left(\ell_{S_a}(p) + 2\,\ell_{T_{a,b}}(p) + 2\,\ell_{T_{b,a}}(p) + \ell_{S_b}(p)\right) && \text{(by the definition of } \mathcal{D}\text{)}\\
&\geq \frac{1}{6} \cdot (3\,\|pa\| + 2\,\|mp\| + \|pb\|) && \text{(by the discussion above)}\\
&= \frac{1}{6} \cdot (3\,\|pa\| + \|pa\|) && \text{(by (4))}\\
&= \frac{2}{3} \cdot \|pa\|,
\end{aligned}$$

---

efficiently. Thus, we can obtain an algorithm that runs efficiently on a Turing machine, provided that we are willing to incur a small additive loss in the approximation factor.

[2] In particular, a is connected to b. If m happens to belong to $\mathcal{P}$, it can be connected to either a or b.

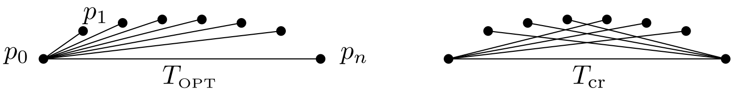

Fig. 2. The point set $\mathcal{P}_n$ consisting of $n+1$ points with equally spaced x-coordinates $0, \ldots, n$, with its longest plane and crossing spanning trees.

Similarly, if p lies to the left of m, we get $\mathbf{E}_{T\sim\mathcal{D}}[\ell_T(p)] \geq (2/3) \cdot \|pb\|$, Altogether, we use (3) to obtain

$$\mathbf{E}_{T\sim\mathcal{D}}[\ell_T(p)] \geq \frac{2}{3} \cdot \max\{\|pa\|, \|pb\|\} = \frac{2}{3} \cdot \ell_{T_{\mathrm{cr}}}(p), \tag{5}$$

for all $p \in \mathcal{P} \setminus \{a\}$, Now, we can put everything together:

$$\begin{aligned} \mathbf{E}_{T\sim\mathcal{D}}[|T|] &= \mathbf{E}_{T\sim\mathcal{D}}\left[\sum_{p\in\mathcal{P}\setminus\{a\}} \ell_T(p)\right] && \text{(by the definition of } \ell_T(p)\text{)} \\ &= \sum_{p\in\mathcal{P}\setminus\{a\}} \mathbf{E}_{T\sim\mathcal{D}}\left[\ell_T(p)\right] && \text{(by linearity of expectation)} \\ &\geq \sum_{p\in\mathcal{P}\setminus\{a\}} \frac{2}{3} \cdot \ell_{T_{\mathrm{cr}}}(p) = \frac{2}{3}\mathrm{OPT}_{\mathrm{cr}}. && \text{(by (5) and the definition of } \ell_{Tcr}(p)\text{)} \end{aligned}$$

Thus, we have proved (2), and the theorem follows. □

In fact, one can find an example where the constant 2/3 is asymptotically tight when comparing to $\mathrm{OPT}_{\mathrm{cr}}$:

Lemma 2.2. *There is a sequence of point sets $\mathcal{P}_1, \mathcal{P}_2, \ldots$ with $|\mathcal{P}_n| = n+1$ and*

$$\lim_{n\to\infty} \frac{\mathrm{OPT}(\mathcal{P}_n)}{\mathrm{OPT}_{\mathrm{cr}}(\mathcal{P}_n)} \leq \frac{2}{3}.$$

Proof. For $n \geq 1$, the set $\mathcal{P}_n = \{p_0, p_1, \ldots, p_n\}$ is the flat point set where the x-coordinate of p_i is i, for $i = 0, \ldots, n$, and all p_i are on the upper convex hull of $\mathcal{P}_n$, see Figure 2.

First, we argue by induction on n that the star S_{p_0} rooted at p_0 is a longest plane spanning tree for $\mathcal{P}_n$, for any $n \geq 1$. For $n = 1$, this is clear, since S_{p_0} is the only spanning tree for $\mathcal{P}_1$.

Now, suppose that $n \geq 2$. Then, the inductive hypothesis states that the star S'_{p_0} on the point set $\mathcal{P}_{n-1}$ rooted at p_0 is a longest plane spanning tree for $\mathcal{P}_{n-1}$.

To perform the inductive step, we first observe that any longest plane spanning tree for $\mathcal{P}_n$ must contain the edge p_0p_n. To see this, let T be a plane spanning tree on $\mathcal{P}_n$ that does not contain the edge p_0p_n. As p_0p_n is the longest edge in $\mathcal{P}_n$, it is also the longest edge on the fundamental cycle in $T \cup \{p_0p_n\}$. Furthermore, as p_0p_n does not cross any other edge, removing any other edge from the cycle gives a plane tree that is longer than T.

We also claim that in any longest plane spanning tree for $\mathcal{P}_n$, either p_0 or p_n must be a leaf. Indeed, consider a plane spanning tree T on $\mathcal{P}_n$ that contains the edge p_0p_n and that has the additional edges p_0p_i and p_jp_n, for $0 < i, j < n$. Since T is plane, we must have $i < j$. Pick the points p_i and p_j such that i is the maximal index and j the minimal index of points with this property. We can now strictly increase the length of T by replacing p_0p_i with p_ip_n (if $i < n/2$) or by replacing p_jp_n by p_0p_j (if $j > n/2$). To see that the new tree is plane, assume without loss of generality that $i < n/2$. As i is the largest index such that the edge p_0p_i exists, the edge p_ip_n does not cross any other edges incident to p_0. As T is plane, the points $p_1, \ldots, p_{i-1}$ are only connected to other points in $p_1 \dots, p_{i-1}$, and these edges are not crossing p_0p_i and thus also do not cross p_ip_n. The fact that i is the maximal index for an edge p_0p_i implies that p_ip_n also does not intersect any edge incident to p_0. Furthermore, all points $p_{i+1}, \dots, p_n$ are incident only to other points in $p_{i+1}, \dots, p_n$ as the edge would otherwise cross p_0p_i. By the convex position of the point set, this directly implies that p_np_i does not intersect any other edges and thus the resulting tree is plane.

By symmetry, we can thus conclude that there is a longest plane spanning tree for $\mathcal{P}_n$ that contains the edge p_0p_n and in which p_n is a leaf. Since p_0p_n does not cross any other edge on $\mathcal{P}_n$, it follows that we can obtain a longest spanning tree for $\mathcal{P}_n$ by taking a longest spanning tree for $\mathcal{P}_{n-1}$ and by adding the edge p_0p_n. By the inductive hypothesis, the star S'_{p_0} on the point set $\mathcal{P}_{n-1}$ is a longest plane spanning tree for $\mathcal{P}_{n-1}$. Hence, it follows that S_{p_0} is a longest plane spanning tree for $\mathcal{P}_n$. Thus, we get

$$\mathrm{OPT}(\mathcal{P}_n) = |S_{p_0}| = \sum_{i=1}^{n} i = \frac{n^2}{2} + \frac{n}{2}.$$

Consider now the tree T_{cr} where all points p_i with $0 \le i \le \lfloor n/2 \rfloor$ are connected to p_n and all points p_i with $\lfloor n/2 \rfloor + 1 \le i \le n$ are connected to p_0 (in particular, T_{cr} contains the edge p_0p_n). This tree is illustrated in Figure 2 on the right. Similar as in the proof of Theorem 2.1, the tree T_{cr} is a longest (crossing) spanning tree for $\mathcal{P}_n$, because every edge p_ip_j that does not appear in T_{cr} is a shortest edge in the fundamental cycle that p_ip_j makes with T_{cr}. Thus, a straightforward summation gives a lower bound on $\mathrm{OPT}_{\mathrm{cr}}(\mathcal{P}_n)$:

$$\begin{aligned}
\mathrm{OPT}_{\mathrm{cr}}(\mathcal{P}_n) &= \sum_{i=1}^{\lfloor n/2 \rfloor} (n-i) + \sum_{i=\lfloor n/2 \rfloor+1}^{n} i && \text{(by the structure of } T_{\mathrm{cr}}\text{)}\\
&= \sum_{i=1}^{\lfloor n/2 \rfloor} \left(\left\lceil \frac{n}{2} \right\rceil - 1 + i \right) + \sum_{i=\lfloor n/2 \rfloor+1}^{n} i && \text{(index transformation } i \mapsto \left\lfloor \frac{n}{2} \right\rfloor + 1 - i\text{)}\\
&= \left\lfloor \frac{n}{2} \right\rfloor \cdot \left(\left\lceil \frac{n}{2} \right\rceil - 1 \right) + \sum_{i=1}^{n} i && \text{(rearranging terms)}\\
&= \frac{n^2}{2} + \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lceil \frac{n}{2} \right\rceil + \frac{n}{2} - \left\lfloor \frac{n}{2} \right\rfloor && \text{(evaluating the sum, rearranging terms)}\\
&\ge \frac{3n^2}{4},
\end{aligned}$$

where the last inequality is due to a case distinction: if n is even, we have

$$\begin{aligned}
\frac{n^2}{2} + \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lceil \frac{n}{2} \right\rceil + \frac{n}{2} - \left\lfloor \frac{n}{2} \right\rfloor &= \frac{n^2}{2} + \frac{n}{2} \cdot \frac{n}{2} + \frac{n}{2} - \frac{n}{2}\\
&= \frac{3n^2}{4},
\end{aligned}$$

and if n is odd, we have

$$\begin{aligned}
\frac{n^2}{2} + \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lceil \frac{n}{2} \right\rceil + \frac{n}{2} - \left\lfloor \frac{n}{2} \right\rfloor &= \frac{n^2}{2} + \left(\frac{n}{2} - \frac{1}{2} \right) \cdot \left(\frac{n}{2} + \frac{1}{2} \right) + \frac{n}{2} - \left(\frac{n}{2} - \frac{1}{2} \right)\\
&= \frac{n^2}{2} + \frac{n^2}{4} - \frac{1}{4} + \frac{1}{2}\\
&= \frac{3n^2}{4} + \frac{1}{4}.
\end{aligned}$$

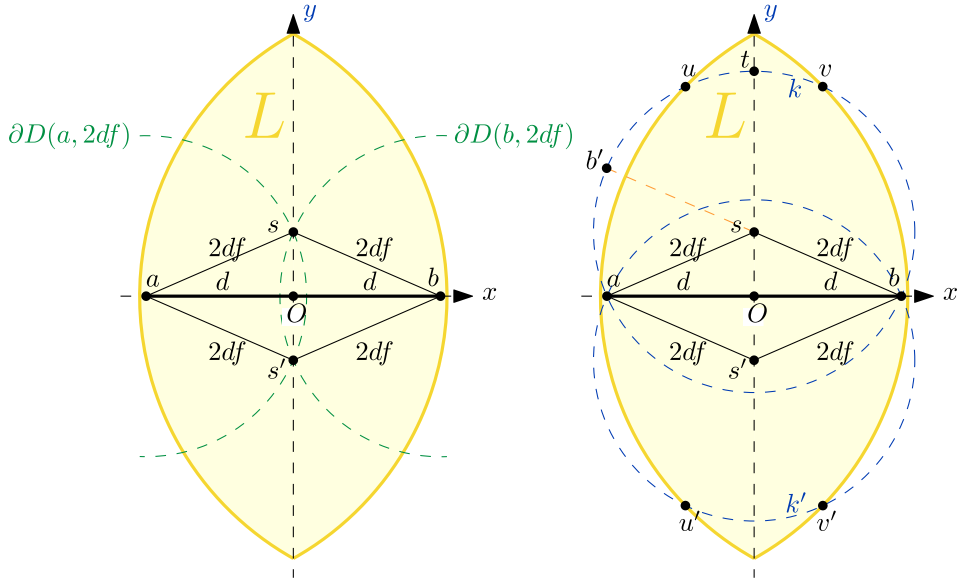

Fig. 3. (left) The points a and b define a longest edge of T_{OPT}. The points s and s' are at distance $2df$ from both a and b. The lens L is defined by a and b with radius 2. (right) The points u, v, u', and v' are at the intersection of the circles with radius $2df$ around s and s' and the boundary of L.

Combining the bounds for $\text{OPT}(\mathcal{P}_n)$ and $\text{OPT}_{\text{cr}}(\mathcal{P}_n)$, we get the desired result:

$$\lim_{n\to\infty} \frac{\text{OPT}(\mathcal{P}_n)}{\text{OPT}_{\text{cr}}(\mathcal{P}_n)} \le \lim_{n\to\infty} \frac{n^2/2 + n/2}{3n^2/4} = \frac{2}{3}.$$

Now we use a similar approach to show the main theorem of this section:

THEOREM 1.1. *For any finite point set $\mathcal{P}$ in general position (the set $\mathcal{P}$ contains no three collinear points), we can compute in polynomial time a plane tree of Euclidean length at least $f \cdot \text{OPT}$, where* OPT *denotes the length of a longest plane tree on $\mathcal{P}$ and $f > 0.5467$ is the fourth smallest real root of the polynomial*

$$P(x) = -80 + 128x + 504x^2 - 768x^3 - 845x^4 + 1096x^5 + 256x^6.$$

PROOF. We outline the proof strategy, referring to lemmas that will be proved later in this section. Without loss of generality, suppose that $\mathcal{P}$ has diameter 2, and let $x, y \in \mathcal{P}$ be two points realizing the diameter of $\mathcal{P}$. Next, fix a longest plane spanning tree T_{OPT} for $\mathcal{P}$, and consider a longest edge ab of T_{OPT}. Note that ab does not necessarily realize the diameter of $\mathcal{P}$, and thus a, b and x, y may differ. Write $\|ab\| = 2d$. By our assumption, we have $d \le 1$. Now, if $2d \le 1/f$, we can use a technique of Alon, Rajagopalan, and Suri [1] to show that one of S_x or S_y is long enough for the desired approximation (see Lemma 2.3). Thus, we will assume from now on that $2df > 1$.

We define several regions in the plane that we use to analyze $\mathcal{P}$ and to derive the desired approximation, refer to Figure 3 for an illustration. First, we choose a coordinate system such that a and b lie on the x-axis and are symmetric with respect to the y-axis, with a on the left. In other words, since $\|ab\| = 2d$, we choose the coordinates such that $a = (-d, 0)$ and $b = (d, 0)$. Since $d \le 1$ and $2df > 1$, the two circles $\partial D(a, 2df)$ and $\partial D(b, 2df)$ have exactly two intersection points. By symmetry, these intersection points lie on the y-axis, one above and one below the x-axis. We let s

be the intersection point that is above the x-axis, and s' the intersection point below the x-axis. By definition, s and s' have distance $2df$ from both a and b.

Now, we define the *lens* $L = D(a, 2) \cap D(b, 2)$. Since $\mathcal{P}$ has diameter 2, we have $\mathcal{P} \subset L$, and since $2df \le 2$ (by our choice of f), we have $s, s' \in L$. Next, consider the circles $k = \partial D(s, 2df)$ and $k' = \partial D(s', 2df)$. We argue that k intersects the boundary ∂L of the lens L. Indeed, the point b' that is symmetric to b with respect to s lies outside L, because $\|bb'\| = 2 \cdot 2df > 2$. On the other hand, the point t on k that is vertically above s on the y-axis lies inside L, because it has y-coordinate

$$\begin{aligned}\sqrt{4d^2f^2 - d^2} + 2df &\le \sqrt{4f^2 - 1} + 2f \\ &\le \sqrt{4 \cdot 0.55^2 - 1} + 2 \cdot 0.55 \\ &\le 1.56 \le \sqrt{3} \le \sqrt{4 - d^2},\end{aligned}$$

as $d \le 1$ and by our choice of $f \le 0.55$. Thus, along the boundary of k, there has to be exactly one intersection point between k and ∂L that lies clockwise between b' and t.[3] We call this intersection point u. Symmetrically, we define v as the analogous intersection between k and ∂L in the upper right quadrant, and u', v' as the analogous intersections between k' and ∂L in the halfplane below the x-axis. Now, we claim that u lies to the left of the y-axis and to the right of a. Indeed, we argued that u lies between the points b' and t, as defined above. Since t lies on the y-axis, the first part of the claim follows. Furthermore, since b' is symmetric to b with respect to s, which lies on the y-axis, and since a is symmetric to b with respect to the y-axis, it follows that b' lies vertically above a, and hence u also lies to the right of a. By symmetry, analogous statements hold for u', v, and v', and we have

$$a_x \le u_x = u'_x \le 0 \le v_x = v'_x \le b_x. \tag{6}$$

We use the points u, v, u' and v' to partition the lens into two regions, the *far region* and the *truncated lens*. The truncated lens is the region of L that lies below the arc of k between u and v in clockwise direction and above the arc of k' between u' and v' in counter-clockwise direction. The remaining parts of L constitute the far region, see Figure 4.

In Lemma 2.4, we show that for any point c in the far region, the triangle abc has acute angles and circumradius $R \ge 2df$. This allows us to argue that if $\mathcal{P}$ contains a point c in the far region, then one of the three stars S_a, S_b, or S_c is long enough for the desired approximation (Lemma 2.4).

It remains to consider the situation that $\mathcal{P}$ lies in the truncated lens. We claim that in this case, one of the trees S_a, $T_{a,b}$, $T_{b,a}$, or S_b yields the desired approximation. For this, we proceed as in the proof of Theorem 2.1. Our goal is to show

$$\max\{|S_a|, |T_{a,b}|, |T_{b,a}|, |S_b|\} \ge f \cdot \mathrm{OPT}. \tag{7}$$

To do this, we fix a parameter $\beta \in (0, 1/2)$, and we define a probability distribution $\mathcal{D}_\beta$ on $\{S_a, T_{a,b}, T_{b,a}, S_b\}$ by

$$\Pr_{\mathcal{D}_\beta}[S_a] = \Pr_{\mathcal{D}_\beta}[S_b] = 1/2 - \beta \text{ and } \Pr_{\mathcal{D}_\beta}[T_{a,b}] = \Pr_{\mathcal{D}_\beta}[T_{b,a}] = \beta.$$

Now it suffices to show that there exists a choice of β such that

$$\mathbf{E}_{T \sim \mathcal{D}_\beta}[|T|] \ge f \cdot \mathrm{OPT}. \tag{8}$$

For this, we consider the edges individually and use linearity of expectation. For $p \in \mathcal{P} \setminus \{a\}$ and $T \in \{S_a, T_{a,b}, T_{b,a}, S_b, T_{\mathrm{OPT}}\}$, let $\ell_T(p)$ be the length of the first edge on the path from p to a in T.

---

[3] Note that two circles intersect in at most two points.

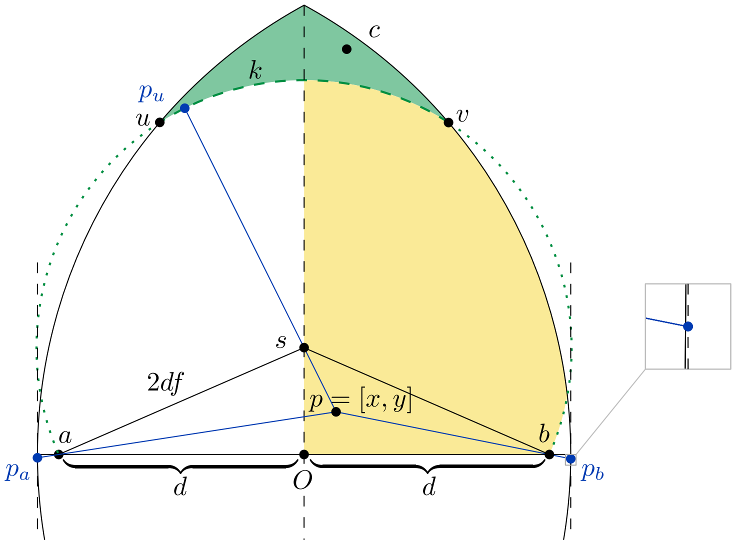

Fig. 4. The lens L is split into the far region (green) and the truncated lens.

Then, we have

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[|T|] = \mathbf{E}_{T\sim\mathcal{D}_\beta}\left[\sum_{p\in\mathcal{P}\setminus\{a\}} \ell_T(p)\right] = \sum_{p\in\mathcal{P}\setminus\{a\}} \mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)]\,. \tag{9}$$

To finish the argument, we show that for any point $p \in \mathcal{P} \setminus \{a\}$, we have

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \geq f \cdot \ell_{T_{\text{OPT}}}(p). \tag{10}$$

In contrast to the proof of Theorem 2.1, this now requires much more work. Below, we will see that $\beta \doteq 0.1604$ gives the desired result. Then, we get

$$\sum_{p\in\mathcal{P}\setminus\{a\}} \mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \geq \sum_{p\in\mathcal{P}\setminus\{a\}} f \cdot \ell_{T_{\text{OPT}}}(p) = f \cdot |T_{\text{OPT}}| = f \cdot \text{OPT}. \tag{11}$$

Thus at least one of the four trees S_a, $T_{a,b}$, $T_{b,a}$, S_b has total length at least $f \cdot \text{OPT}$.

We outline the proof of (10), again referring to lemmas that will be stated and proved below. If $p = b$, then (10) is immediate, because the five trees S_a, $T_{a,b}$, $T_{b,a}$, S_b, T_{OPT} all contain the edge ab. Thus, let $p \in \mathcal{P} \setminus \{a, b\}$, and assume without loss of generality that p is in the upper right quadrant, i.e., that $p = (x, y)$ with $x, y \geq 0$. First, we establish an upper bound on $\ell_{T_{\text{OPT}}}(p)$, by estimating the length of a longest possible line segment that (i) has p as an endpoint, (ii) lies in L, and (iii) does not cross the edge ab. For this, we identify three possible candidates for p_a, p_b, and p_u of the other endpoint (see Figure 4 for an illustration). The first two points p_a and p_b provide upper bounds for the case that the other endpoint of the longest possible edge lies below the x-axis: the point p_a is the point with x-coordinate $-(2-d)$ on the ray pa; and for p_b, there are two cases: if $x < d$, then p_b is the point with x-coordinate $2-d$ on the ray pb; if $x \geq d$, the ray pb does not intersect the vertical line $x = 2-d$, and we set $p_b = b$. The third point p_u is the furthest point from p on the part of the boundary of the far region that is on the circle $k = \partial D(s, 2df)$. The proof now proceeds in the following steps:

(1) In Lemma 2.5, we establish the upper bound

$$\ell_{T_{\text{OPT}}}(p) \le \min\left\{2d,\ \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\right\}, \tag{12}$$

estimating the length of the longest possible line segment in the lens L with endpoint p that does not cross ab, and using the fact that all edges in T_{OPT} have length at most $2d$. Thus, it suffices to compare $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)]$ with $\|pp_a\|$, $\|pp_b\|$, and $\|pp_u\|$.

(2) In Lemma 2.6, we show that $\|pp_a\|$ can never be smaller than $\|pp_b\|$, and thus the term $\|pp_b\|$ in (12) can be dropped. That is, it suffices to compare $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)]$ with $\|pp_a\|$ and $\|pp_u\|$.

(3) In Lemma 2.7, we establish a lower bound on $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)]$, by analyzing how p is connected in the trees S_a, $T_{a,b}$, $T_{b,a}$, and S_b.

(4) In Lemmas 2.8 and 2.9, we relate the lower bound from Lemma 2.7 to $\|pp_a\|$ and $\|pp_u\|$. In this way, we identify constraints on β such that

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge f\cdot\min\{2d, \|pp_a\|\} \text{ and}$$
$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge f\cdot\min\{2d, \|pp_u\|\}.$$

It remains to find a $\beta \in (0, 1/2)$ that satisfies all constraints from Lemmas 2.8 and 2.9. The most stringent constraints turn out to be those from Lemma 2.9, stating that

$$\frac{2f-1}{2\sqrt{5-8f}-1} \le \beta \le 1 - f\sqrt{4f^2-1} - 2f^2. \tag{13}$$

In Lemma 2.10, we use straightforward algebra to show that our choice of f makes the left-hand side and the right-hand side of (13) equal. Hence, we set $\beta \doteq 0.1604$, their joint value. (The value $f = 5/8$ also makes both sides of (13) equal, but it would give $\beta < 0$.)

To summarize, for the approximation guarantee $f \doteq 0.5467$ given in the theorem, the value $\beta = 1 - f\sqrt{4f^2-1} - 2f^2 \doteq 0.1604$, and for every point $p \in \mathcal{P} \setminus \{a\}$, we have

$$\begin{aligned}\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] &\ge f\cdot\max\left\{\min\{2d, \|pp_a\|\}, \min\{2d, \|pp_u\|\}\right\}\\ &\ge f\cdot\min\left\{2d,\ \max\{\|pp_a\|, \|pp_u\|\}\right\}\\ &\ge f\cdot\ell_{T_{\text{OPT}}}(p),\end{aligned}$$

and the result follows. □

It remains to prove Lemmas 2.3 to 2.10. Their statements rely on the notation introduced in the proof outline of Theorem 1.1, so we recommend to first consult the paragraphs above.

Lemma 2.3. *Let $x, y \in \mathcal{P}$ be two points that realize the diameter of $\mathcal{P}$. Suppose that $\|xy\| = 2$ and that all edges of the optimal tree T_{OPT} have length at most $1/f$. Then, we have* $\max\{|S_x|, |S_y|\} \ge f\cdot\text{OPT}$.

Proof. By the triangle inequality, for any point $p \in \mathcal{P}$, we have $\|xp\| + \|yp\| \ge \|xy\| = 2$. Hence, the total length of the stars S_x and S_y is

$$|S_x| + |S_y| = \sum_{p\in\mathcal{P}} (\|xp\| + \|yp\|) \ge n\cdot\|xy\| = 2n.$$

On the other hand, since each of the $n-1$ edges in T_{OPT} has length at most $1/f$, we get

$$\max\{|S_x|, |S_y|\} \ge \frac{|S_x| + |S_y|}{2} \ge n > f\cdot(n-1)\cdot\frac{1}{f} \ge f\cdot\text{OPT},$$

and we are done. □

Lemma 2.4. *Let ab (with $\|ab\| = 2d$) be a longest edge of T_{OPT}. If $\mathcal{P}$ contains a point c in the far region, then* $\max\{|S_a|, |S_b|, |S_c|\} \ge f\cdot\text{OPT}$.

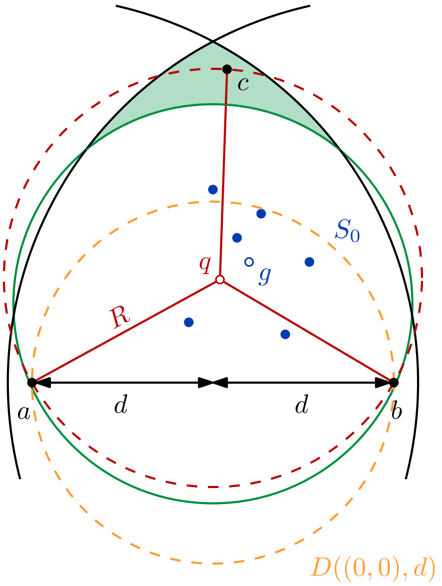

Fig. 5. We can chose $v = a$. The orange disk does not intersect the far region above the x-axis. The circumradius of the triangle abc is marked red.

Proof. First, we argue that the triangle abc is acute: by (6), the far region lies vertically between a and b, and thus the angles at a and b in the triangle abc are at most $\pi/2$. Furthermore, the upper half of the disk $D((0,0),d)$ is completely contained in k (Figure 5). Thus, since c lies vertically above k, and hence vertically above $D((0,0),d)$, Thales's theorem implies that the angle at c is also at most $\pi/2$. It follows that all three angles are at most $\pi/2$ and the triangle abc is acute.

Next, we argue that the circumradius R of the triangle abc satisfies $R \geq 2df$: Indeed, since the center of the circumcircle of a triangle lies at the intersection of the perpendicular bisectors of its sides, the center of the circumcircle of abc lies on the y-axis. Consider the point c' that lies vertically below c on k. The triangle abc' is acute and has k as its circumcircle, i.e., its circumradius is $2df$. Now, if we move c' vertically towards c, the intersection of the perpendicular bisector of ac with the y-axis moves upward, and hence its distance to a increases. Thus, the circumradius R of abc is as least as large as the circumradius of abc', and we have $R \geq 2df$.

Now, set $\mathcal{P}_0 = \mathcal{P} \setminus \{a,b,c\}$, and let $g = (1/|\mathcal{P}_0|) \cdot \sum_{p \in \mathcal{P}_0} p$ be the corresponding center of mass.[4] By the definition of g, we have that $\sum_{p \in \mathcal{P}_0} \overrightarrow{vp} = |\mathcal{P}_0| \cdot \overrightarrow{vg}$, for every point $v \in \mathbb{R}^2$, where $\overrightarrow{ab}$ denotes the two-dimensional vector that is defined by a and b.

By the triangle inequality, this implies that for every $p \in \mathbb{R}^2$, we have

$$\sum_{p \in \mathcal{P}_0} \|vp\| \geq |\mathcal{P}_0| \cdot \|vg\| = (n-3)\,\|vg\|.$$

Since the triangle abc is acute, Thales's theorem implies that the circumcenter q of abc lies in its interior or on its boundary. Thus, if we consider the rays qa, qb, and qc from the circumcenter q through a, b, and c, the angle between two consecutive rays is as most π. Hence, two of these angles have to be at least $\pi/2$. Wlog, suppose that the angle between qc and qa and the angle between qc and qb are both at least $\pi/2$. Then, the tangent line for the disk $D(c,R)$ through q separates a and

---

[4] Note that $\mathcal{P}_0$t may have points below and above the x-axis, but that does not affect the argument.

b from c. But then, the interiors of the disks $D(a,R)$, $D(b,R)$, and $D(c,R)$ cannot have a common point, because the tangent separates the interior of $D(a,R)\cap D(b,R)$ from the interior of $D(c,R)$. It follows that there exists a vertex $v\in\{a,b,c\}$ such that $\|vg\|\geq R$.

Furthermore, the acuteness of abc implies that $\|va\|+\|vb\|+\|vc\|\geq 2R$. Indeed, suppose wlog that $v=a$. Let q be the circumcenter of abc. We have $2R=\|bq\|+\|qc\|$. We already saw that the circumcenter q lies inside abc, so let b' be the intersection of the ray bq with the edge ac. Applying the triangle inequality in the triangle $qb'c$, we get $\|qc\|\leq\|qb'\|+\|b'c\|$, and applying the triangle inequality in the triangle $bb'a$, we get $\|bb'\|\leq\|ab\|+\|ab'\|$. Thus,

$$2R=\|bq\|+\|qc\|\leq\|bq\|+\|qb'\|+\|b'c\|=\|bb'\|+\|b'c\|\leq\|ab\|+\|ab'\|+\|b'c\|=\|ab\|+\|ac\|,$$

as claimed.

Putting everything together, we get

$$\begin{aligned}|S_v| &= \sum_{p\in\mathcal{P}}\|vp\|\\ &= \sum_{p\in\mathcal{P}_0}\|vp\|+\|va\|+\|vb\|+\|vc\|\\ &\geq (n-3)\cdot\|vg\|+2R\\ &\geq (n-3)\cdot R+2R\\ &= (n-1)\cdot R\\ &\geq (n-1)\cdot 2df\\ &\geq f\cdot \mathrm{OPT},\end{aligned}$$

where in the last inequality we used that every edge of T_{OPT} has length at most $2d$. □

Recall that ab is the longest edge of T_{OPT} and that we assumed $a=(-d,0)$, $b=(d,0)$ and $2df>1$. Furthermore, recall that The point s has coordinates $s=(0,\sqrt{(2df)^2-d^2})$. We also noted that the circle $k=D(s,\|sa\|)$ always intersects the lens $D(a,2)\cap D(b,2)$, and we used this to define the intersection points u and v, as in Figure 3.

For each point $p=(x,y)$ in the truncated lens with $x,y\geq 0$, recall that we have defined the following points:

- p_a: the point on the ray pa whose x-coordinate equals $-(2-d)$;
- p_b: the point on the ray pb whose x-coordinate equals $2-d$, if $x(p)<d$, and $p_b=b$, if $x(p)\geq d$; and
- p_u: the point on the arc of k from u to v that is furthest from p. Thus, if ray ps intersects the arc of k from u to v, then p_u is that intersection point, and therewise $p_u=u$.

Now, we show that these three special points p_a, p_b, and p_u suffice to obtain an upper bound for $\ell_{T_{\mathrm{OPT}}}(p)$.

Lemma 2.5. *For every point $p=(x,y)$ in the truncated lens with $x,y\geq 0$, we have*

$$\ell_{T_{\mathrm{OPT}}}(p)\leq\min\left\{2d,\max\{\|pp_a\|,\|pp_b\|,\|pp_u\|\}\right\}.$$

Proof. As p lies in the upper right quadrant, we have to consider only the truncated lens in this quadrant. Let l and r be the left- and rightmost points of $D(a,2)\cap D(b,2)$, i.e., let $r=(2-d,0)$ and $l=(d-2,0)$.

We further subdivide the truncated lens into regions; see Figure 6 for an illustration: (i) the region E lies inside the truncated lens, but outside of $D(l,2d)$; (ii) the region N lies in the intersection of

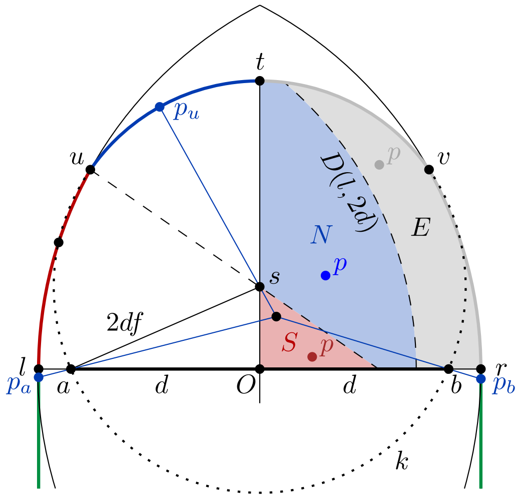

Fig. 6. For $p \in E$, we have $\ell_{T_{\text{OPT}}}(p) \leq 2d$, for $p \in N$, we have $\ell_{T_{\text{OPT}}}(p) \leq \max\{\|pp_a\|, \|pp_u\|\}$, and for $p \in S$, we have $\ell_{T_{\text{OPT}}}(p) \leq \max\{\|pp_a\|, \|pp_u\|\}$.

the truncated lens and $D(l, 2d)$, and above the line through u and s; and (iii) the region S lies in the intersection of the truncated lens and $D(l, 2d)$, and below the line though u and s.

First, consider the case that $p \in E$. Then, we have $\|pp_a\| \geq \|pl\| \geq 2d$. Hence, it follows that

$$\max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\} \geq 2d,$$

and thus

$$\min\{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\} = 2d.$$

Since by our definition of d, we have $\ell_{T_{\text{OPT}}}(p) \leq 2d$ for every $p \in \mathcal{P}$, the claim now follows for the case $p \in E$.

Next, consider the case that $p \in N \cup S$. Now, since p must be to the left of b, it follows that $x(p_b) = 2 - d$. Let p_f be the point within the truncated lens that is furthest from p and can be connected to p by a line segment that does not intersect ab. We assume that there is no point from $\mathcal{P}$ in the far region, so we must have $\ell_{T_{\text{OPT}}}(p) \leq \|pp_f\|$. Since the truncated lens is bounded, p_f lies on the boundary of the truncated lens. Since $\ell_{T_{\text{OPT}}}(p) \leq 2d$ by our definition of d, it suffices to show that $\|pp_f\| \leq \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}$. We distinguish four cases, depending on the quadrant containing p_f:

(1) p_f lies in the upper right quadrant: consider the reflection p'_f of p_f about the y-axis. Since $x \geq 0$, we have $\|pp'_f\| \geq \|pp_f\|$, and the inequality is strict for $x \geq 0$. Thus, if $x > 0$, this case cannot occur, and if $x = 0$, it is handled in the second case

(2) p_f lies in the left quadrant: let t be the top-most point of the truncated lens. We distinguish two subcases:
   (a) p_f lies on the arc tu (Figure 6): if p lies in N, then for any q on the arc tu, we have $\|pq\| \leq \|pu\| = \|pp_u\|$, thus we also get $\|pp_f\| \leq \|pp_u\|$. If p lies in S, then, by the triangle inequality, we get $\|pp_f\| \leq \|ps\| + \|sp_f\| = \|ps\| + 2df = \|pp_u\|$. Thus, in either situation, we have $\|pp_f\| \leq \|pp_u\|$.
   (b) p_f lies on the arc ul: we claim that now, it holds that $\|pp_f\| \leq \max\{\|pl\|, \|pu\|\}$. Indeed, since l, p_f, and u all lie on $\partial D(b, 2)$, the perpendicular bisectors of the segments $p_f l$ and

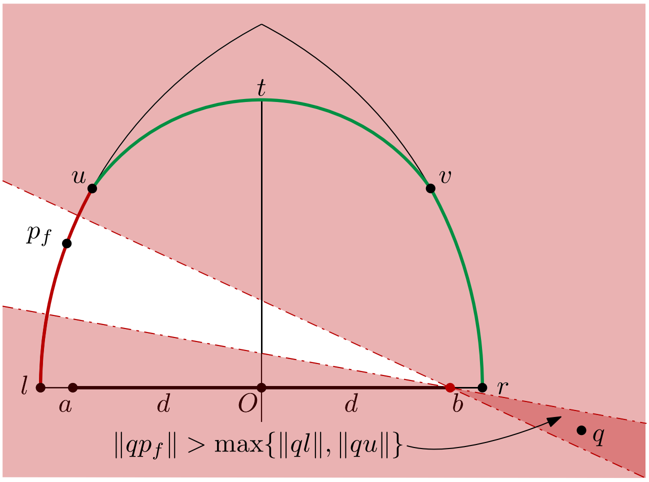

Fig. 7. The point p_f lies on the arc ul. The set of points q such that $\|qp_f\| > \max\{\|ql\|, \|qu\|\}$ forms a convex wedge with vertex b fully contained in the fourth quadrant.

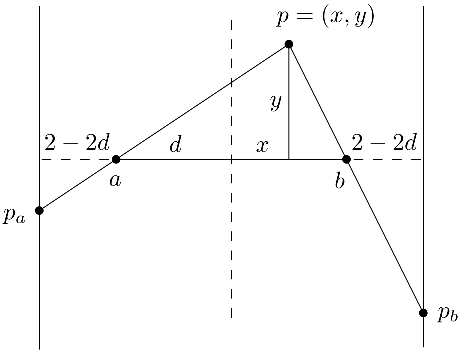

Fig. 8. The situation for Lemma 2.6.

p_fu intersect at b, and thus the points q for which $\|qp_f\| > \max\{\|ql\|, \|qu\|\}$ lie in a convex cone with vertex b that is fully contained in the lower right quadrant, see Figure 7. Now, since $\|pl\| \leq \|pp_a\|$ and $\|pu\| \leq \|pp_u\|$, we get $\|pp_f\| \leq \max\{\|pp_a\|, \|pp_u\|\}$ as desired.

(3) p_f lies in the lower left quadrant: then, we have $\|pp_f\| \leq \|pp_a\|$, by the definition of p_a.
(4) p_f lies in the lower right quadrant: then, we have $\|pp_f\| \leq \|pp_b\|$, by the definition of p_b. ☐

Now we show, that if p lies in the first quadrant, the point p_b can be ignored.

Lemma 2.6. *For every point $p = (x, y)$ with $x, y \geq 0$ in the truncated lens, we have that if $\|pp_a\| \leq 2d$, then $\|pp_b\| \leq \|pp_a\|$.*

Proof. We present a direct analytic proof (we are not aware of an argument that uses only elementary geometry). First, if $x = 0$, the claim follows from symmetry, since then we have $\|pp_b\| = \|pp_a\|$. Thus, we consider only the case $x > 0$. Since $\|pp_a\| \leq 2d$, we must have $x < d$, and therefore p_b has x-coordinate $2 - d$. See Figure 8.

Using similar triangles, we have

$$\|pp_a\| = \|pa\| \cdot \frac{2-d+x}{d+x}, \tag{14}$$

and using the Pythagorean theorem, we have $\|pa\|^2 = y^2 + (d+x)^2$. Thus, we have

$$\|pp_a\|^2 = (y^2 + (d+x)^2) \cdot \frac{(2-d+x)^2}{(d+x)^2}. \tag{15}$$

Similarly, using $\|pp_b\| = \|pb\| \cdot \frac{2-d+x}{d-x}$ and $\|pb\|^2 = y^2 + (d-x)^2$, we express $\|pp_b\|^2$ as

$$\|pp_b\|^2 = (y^2 + (d-x)^2) \cdot \frac{(2-d-x)^2}{(d-x)^2}.$$

Thus, our goal $\|pp_b\|^2 \leq \|pp_a\|^2$ can be stated as

$$(y^2 + (d-x)^2) \cdot \frac{(2-d-x)^2}{(d-x)^2} \leq (y^2 + (d+x)^2) \cdot \frac{(2-d+x)^2}{(d+x)^2}.$$

Separating the terms involving y^2 from the rest, this becomes

$$y^2 \cdot \frac{(2-d-x)^2(d+x)^2 - (2-d+x)^2(d-x)^2}{(d+x)^2(d-x)^2} \leq (2-d+x)^2 - (2-d-x)^2. \tag{16}$$

For the right-hand side of (16), we have

$$\begin{aligned}(2-d+x)^2 - (2-d-x)^2 &= (2-d)^2 + x^2 + 2(2-d)x - (2-d)^2 - x^2 + 2(2-d)x \\ &= 4(2-d)x.\end{aligned}$$

For the numerator on the left-hand side of (16), we get

$$\begin{aligned}&(2-d-x)^2(d+x)^2 - (2-d+x)^2(d-x)^2 \\ &= ((2-d)^2 - 2(2-d)x + x^2)(d^2 + 2xd + x^2) - ((2-d)^2 + 2(2-d)x) + x^2)(d^2 - 2xd + x^2) \\ &= 2(2-d)^2 2xd - 4(2-d)xd^2 - 4(2-d)x^3 + 2x^2 2xd \\ &= 4(2-d)^2 xd - 4(2-d)xd^2 - 4(2-d)x^3 + 4x^3 d \\ &= 4x\left[(2-d)^2 d - (2-d)d^2 - (2-d)x^2 + x^2 d\right] \\ &= 4x\left[4d - 4d^2 + d^3 - 2d^2 + d^3 - 2x^2 + dx^2 + dx^2\right] \\ &= 4x\left[4d - 4d^2 + d^3 - 2d^2 + d^3 - 2x^2 + 2dx^2\right] \\ &= 4x\left[4d - 2d^2 - 2x^2 - 4d^2 + 2d^3 + 2dx^2\right] \\ &= 4x \cdot 2(1-d)(2d - d^2 - x^2).\end{aligned}$$

Thus, by plugging back into (16) and dividing by $4x > 0$, the goal $\|pp_b\|^2 \leq \|pp_a\|^2$ becomes

$$y^2 \cdot \frac{2(1-d)(2d - d^2 - x^2)}{(d+x)^2(d-x)^2} \leq 2 - d. \tag{17}$$

Using (15), the assumption $\|pp_a\|^2 \leq (2d)^2$ can be equivalently rewritten as

$$(y^2 + (d+x)^2) \cdot \frac{(2-d+x)^2}{(d+x)^2} \leq 4d^2.$$

Solving for y^2, this gives

$$y^2 \leq \frac{(4d^2 - (2-d+x)^2)(d+x)^2}{(2-d+x)^2}. \tag{18}$$

We plug in the upper bound on y^2 from (18) into (17), cancel the term $(d+x)^2$, and clear the denominators. This leaves us with proving

$$(4d^2 - (2-d+x)^2) \cdot 2(1-d)(2d - d^2 - x^2) \le (2-d+x)^2 \cdot (2-d)(d-x)^2$$

which, upon expanding the parentheses and collecting the terms, becomes

$$0 \le (16d - 32d^2 + 8d^3 + 16d^4 - 7d^5) + x^2(-8d + 16d^2 - 10d^3) + dx^4. \tag{19}$$

For any fixed $d > 0$, the right-hand side $Q(d, x^2)$ is a quadratic function of x^2 with positive coefficient $d > 0$ for the leading term $(x^2)^2$. Hence, the minimum of $Q(d, x^2)$ is attained when

$$x^2 = \frac{8d - 16d^2 + 10d^3}{2d} = 4 - 8d + 5d^2.$$

Plugging $x^2 = 4 - 8d + 5d^2$ into (19) and expanding the parentheses for one last time, we are left to prove

$$0 \le 32d^2 - 96d^3 + 96d^4 - 32d^5 = 32d^2(1 - 3d + 3d^2 - d^3) = 32d^2(1-d)^3,$$

which is true since $d \le 1$. ☐

Now we give a general lower bound on $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)]$ that we will then use in Lemmas 2.8 and 2.9.

Lemma 2.7. *Let $p = (x, y)$ be any point in the plane with $x, y \ge 0$ and let $\beta \in (0, 1/2)$ be a real number. Then*

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge \frac{d \cdot (1-\beta) + x \cdot 2\beta}{d+x} \cdot \|pa\|.$$

Proof. Expanding the definition, we have

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge (1/2 - \beta) \cdot \|pa\| + \beta \cdot \|pa\| + \beta \cdot x + (1/2 - \beta) \cdot \|pb\|.$$

Let $p' = (-x, y)$ be the reflection of p about the y-axis (see Figure 9). By the triangle inequality, $\|p'p\| + \|pb\| \ge \|p'b\| = \|pa\|$. This leads to $x = \frac{1}{2}\|p'p\| \ge \frac{1}{2}(\|pa\| - \|pb\|)$, and we obtain

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge \frac{1}{2} \cdot \|pa\| + \frac{1}{2}\beta \cdot \|pa\| + \left(\frac{1}{2} - \frac{3}{2}\beta\right) \cdot \|pb\|.$$

Next, we claim that $\|pb\| \ge \frac{d-x}{d+x} \cdot \|pa\|$: Indeed, upon squaring, using the Pythagorean theorem and clearing the denominators this becomes $y^2 \cdot 4dx \ge 0$ which is true. Using this bound on the term containing $\|pb\|$, we finally get the desired

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge \frac{(1+\beta)(d+x) + (1-3\beta)(d-x)}{2(d+x)} \cdot \|pa\| = \frac{(1-\beta) \cdot d + 2\beta \cdot x}{d+x} \cdot \|pa\|.$$ ☐

Lemma 2.8. *Let $p = (x, y)$ be any point in the truncated lens with $x, y \ge 0$. Then, if*

$$\frac{2f-1}{5-8f} \le \beta \le \frac{1}{2} \cdot f,$$

we have $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge f \cdot \min\{2d, \|pp_a\|\}$.

Proof. It suffices to show that:

(1) If $x \ge 3d - 2$, then $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge f \cdot 2d$.
(2) If $x \le 3d - 2$, then $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \ge f \cdot \|pp_a\|$.

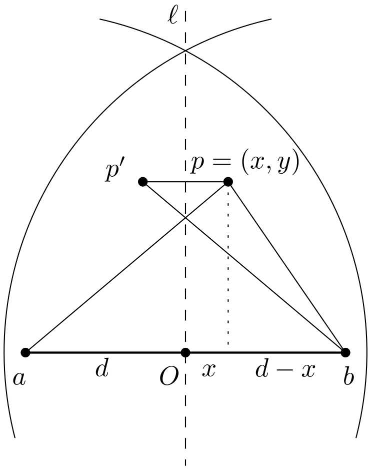

Fig. 9. Mirroring p along the y-axis in Lemma 2.7.

We consider those two cases independently.

(1) Using Lemma 2.7 and the inequalities $\|pa\| \geq d + x$ and $x \geq 3d - 2$, we rewrite

$$\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \geq \frac{d \cdot (1-\beta) + x \cdot 2\beta}{d + x} \cdot \|pa\| \geq d - d\beta + (3d-2) \cdot 2\beta = \beta(5d-4) + d.$$

Hence it suffices to prove $\beta \cdot (5d - 4) \geq d(2f - 1)$. Using the lower bound on β and $4 \leq 8df$ we get

$$\beta \cdot (5d-4) \geq \beta \cdot (5d - 8df) \geq \frac{2f-1}{5-8f} \cdot d \cdot (5-8f) = d(2f-1)$$

as desired. Note that $2f - 1$ and $5 - 8f$ are both positive.

(2) By (14), we have $\|pp_a\| = \|pa\| \cdot \frac{2-d+x}{d+x}$. Using Lemma 2.7, it suffices to prove

$$d(1-\beta) + x \cdot 2\beta \geq f(2-d+x)$$
$$d(1-\beta) - f(2-d) \geq x(f - 2\beta).$$

Since $f \geq 2\beta$ by assumption, the right-hand side is increasing in x and we can plug in $3d - 2$ for x. This leaves us with proving the inequality

$$d(1-\beta) - f(2-d) \geq (3d-2)(f-2\beta)$$
$$\beta \cdot (5d-4) \geq d(2f-1),$$

which is the same inequality as in the first case. ☐

Lemma 2.9. *Let $p = (x, y)$ be any point in the truncated lens with $x, y \geq 0$. Suppose that $\beta < \frac{151}{304} \cdot f$, $\frac{1}{2} \leq f < \frac{19}{32}$, and*

$$\frac{2f-1}{2\sqrt{5-8f}-1} \leq \beta \leq 1 - f\sqrt{4f^2-1} - 2f^2.$$

Then $\mathbf{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)] \geq f \cdot \min\{2d, \|pp_u\|\}$.

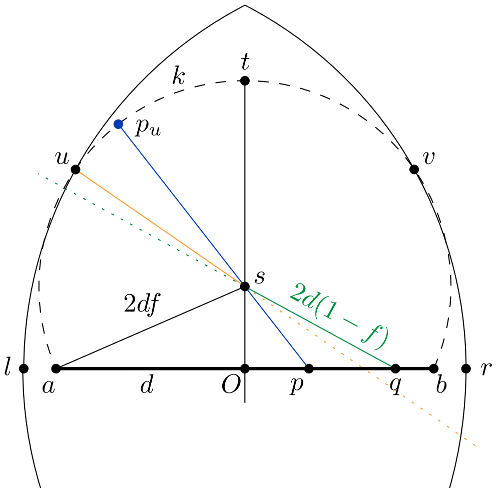

Fig. 10. Lemma 2.9.

Proof. Because of the lower bound for $\mathrm{E}_{T\sim\mathcal{D}_\beta}[\ell_T(p)]$ in Lemma 2.7, to show the statement it suffices to show that

$$\frac{d\cdot(1-\beta)+x\cdot 2\beta}{d+x}\cdot\|pa\|\geq f\cdot\min\{2d,\|pp_u\|\}. \tag{20}$$

We will consider the two cases $y\leq y(u)$ and $y>y(u)$ separately.

Case 1: $y\leq y(u)$ Define $\lambda=\frac{d\cdot(1-\beta)+x\cdot 2\beta}{d+x}\cdot\|pa\|$. We have to show that $\lambda\geq f\cdot\min\{2d,\|pp_u\|\}$. Note that λ is an increasing function in y because $\|pa\|$ is also increasing in y. On the other hand, $\|pp_u\|$ is a decreasing function in y for $y\leq y(u)$: If $p_u\neq u$, then $\|pp_u\|=\|ps\|+2df$ is decreasing in y, and if $p_u=u$ then $\|pp_u\|=\|pu\|$ decreases when y increases (for $y\leq y(u)$). It follows that $\min\{2d,\|pp_u\|\}$ is a decreasing function in y for $y\leq y(u)$.

As $\min\{2d,\|pp_u\|\}$ is decreasing and λ is increasing in y, for $y\leq y(u)$, to handle our current case it suffices to show (20) for $y=0$. Now we have $\|pa\|=d+x$, so we can rewrite λ as $\lambda_0=d\cdot(1-\beta)+x\cdot 2\beta$, which is positive.

Let $q=(q_x,0)$, $q_x\geq 0$ be the point on the x-axis such that $\|qs\|=2d(1-f)$. This means that for $p=q$ we have $\|pp_u\|\leq\|ps\|+\|sp_u\|=2d$ (see Figure 10). We consider subcases depending on whether $0\leq x\leq q_x$ or $x>q_x$.

Case 1a: $0\leq x\leq q_x$. We will show that in this case $\lambda_0\geq f\cdot\|pp_u\|$. The Pythagorean theorem gives

$$f\cdot\|pp_u\|\leq f\cdot\|ps\|+f\cdot\|sp_u\|=f\sqrt{x^2+(2df)^2-d^2}+2df^2$$

Therefore, substituting λ_0, it suffices to show

$$d\cdot(1-\beta)+x\cdot 2\beta\geq f\sqrt{x^2+(2df)^2-d^2}+2df^2. \tag{21}$$

When $\beta<\frac{151}{304}\cdot f$ and $f\leq\frac{19}{32}$, the term $d\cdot(1-\beta)-2df^2$ is positive and the last inequality (21) is equivalent to

$$0\geq f^2\cdot(x^2+(2df)^2-d^2)-(d\cdot(1-\beta)+x\cdot 2\beta-2df^2)^2$$

The right-hand side is a quadratic function of x with positive coefficient $f^2-4\beta^2$ by the leading term, hence it suffices to check the inequality (21) for $x\in\{0,q_x\}$.

For $x = 0$, we need to check that $d(1 - \beta) \geq f\sqrt{d^2(4f^2 - 1)} + 2df^2$, which reduces precisely to the assumption

$$\beta \leq 1 - f\sqrt{4f^2 - 1} - 2f^2.$$

For $x = q_x$, the Pythagorean theorem gives $(q_x)^2 = (2d(1-f))^2 + d^2 - (2df)^2 = d^2 \cdot (5 - 8f)$, hence $q_x = d\sqrt{5 - 8f}$. The point q has been selected so that $\|ps\| + \|sp_u\| = 2d$, and therefore the right side of (21) is $2df$. We thus have to verify that

$$\begin{aligned} d(1 - \beta) + d\sqrt{5 - 8f} \cdot 2\beta &\geq f \cdot 2d \\ \beta \cdot (2\sqrt{5 - 8f} - 1) &\geq 2f - 1. \end{aligned} \tag{22}$$

Since $f < 19/32$, the term in the parentheses on the left-hand side is positive, and after dividing we obtain precisely the assumption.

Case 1b: $q_x < x$. In this case we show that $\lambda_0 \geq f \cdot 2d$. Since the term λ_0 is increasing in x and $2d$ is constant, we only need to show that $\lambda_0 \geq f \cdot 2d$ for $x = q_x$. However, this was already shown in the previous case; see (22).

Case 2: $y > y(p_u)$ In this case we have $\|pp_u\| = \|pu\| \leq \|uv\|$. Furthermore, because $4df > 2d$, the intersection of the line supporting bs with the circle k is outside the lens. This intersection point has x-coordinate $-d$ because of symmetry with respect to s, and since u is above it, we have $x(u) \geq -d$ as well as $x \leq x(v) \leq d$. So we have $\min\{2d, \|pp_u\|\} = \|pp_u\|$. This means that to show (20), it is enough to show

$$\frac{d(1 - \beta) + 2x\beta}{d + x} \cdot \|pa\| \geq f \cdot \|pp_u\|.$$

As p lies above the horizontal line through u, we get $\|pa\| \geq \|pp_u\|$, thus it suffices to show

$$\begin{aligned} \frac{d(1 - \beta) + 2x\beta}{d + x} &\geq f \\ d - d\beta + 2x\beta &\geq fd + fx \\ d(1 - (\beta + f)) &\geq x(f - 2\beta) \\ x &\leq d \cdot \frac{1 - (\beta + f)}{f - 2\beta}, \end{aligned}$$

where we have used that $f - 2\beta > 0$. As we know $x \leq d$, this is true for

$$\begin{aligned} \frac{1 - (\beta + f)}{f - 2\beta} &\geq 1 \\ 1 - (\beta + f) &\geq f - 2\beta \\ \beta &\geq 2f - 1, \end{aligned}$$

where we have again used that $f - 2\beta > 0$. The last condition is a looser bound than the left side of the statement of the lemma as $f \geq \frac{1}{2}$ and therefore $2 \cdot \sqrt{5 - 8f} - 1 \leq 1$. □

For the following lemma, we provide a sketch of how to solve it by hand'. One can also use advanced software for algebraic manipulation.

Lemma 2.10. *The positive solutions of*

$$\frac{2x - 1}{2\sqrt{5 - 8x} - 1} = 1 - x\sqrt{4x^2 - 1} - 2x^2$$

are $x = \frac{5}{8}$ *and the fourth smallest root of*

$$-80 + 128x + 504x^2 - 768x^3 - 845x^4 + 1096x^5 + 256x^6.$$

Proof. Setting the polynomials $q_1(x) = 4x^2 - 1$ and $q_2(x) = 5 - 8x$, and multiplying both sides of the equation by the denominator on the left-hand side, we are left with the equation

$$\begin{aligned} 2x - 2x^2 &= x\sqrt{q_1(x)} + (2 - 4x^2)\sqrt{q_2(x)} - 2x\sqrt{q_1(x)q_2(x)} \\ 2x - 2x^2 + 2x\sqrt{q_1(x)q_2(x)} &= x\sqrt{q_1(x)} + (2 - 4x^2)\sqrt{q_2(x)}. \end{aligned}$$

Squaring both sides of the equation, which may introduce additional roots, we get, for some polynomials $q_3(\,), \ldots, q_6(\,)$, the equation

$$\begin{aligned} q_3(x) + q_4(x)\sqrt{q_1(x)q_2(x)} &= q_5(x) + q_6(x)\sqrt{q_1(x)q_2(x)} \\ q_3(x) - q_5(x) &= (q_6(x) - q_4(x))\sqrt{q_1(x)q_2(x)}. \end{aligned}$$

Squaring both sides gain, which may introduce additional solutions, we get the polynomial

$$8(x - \tfrac{5}{8})(-80 + 128x + 504x^2 - 768x^3 - 845x^4 + 1096x^5 + 256x^6) = 0$$

This polynomial has seven real roots that can be approximated numerically. The smallest three roots of this polynomial are negative ($x \doteq -4.82037$, $x \doteq -0.657898$ and $x \doteq -0.523446$). The fourth smallest root, $x \doteq 0.546723$, is a solution to the original equation. The fifth and sixth roots are $x \doteq 0.577526$ and $x \doteq 0.596211$, which are not solutions to the original equation. The largest root of the polynomial is $x = \frac{5}{8}$, which is also a solution to the original equation. □

3 Point Sets in Convex Position and Flat Point Sets in Convex Position

In this section, we present two results for point sets in convex position. First, we show that if $\mathcal{P}$ is a point set in convex position, then the longest plane tree is a caterpillar (see Theorem 1.2) and that any caterpillar could be the unique longest plane tree (see Theorem 1.3). Second, by looking at suitable flat point sets in convex position, we prove upper bounds on the approximation factor $\mathrm{BoundDiam}(d)$ achieved by the longest plane tree among those with diameter at most d.

3.1 Point Sets in Convex Position and Caterpillars

A tree C is called *caterpillar* if it contains a path P such that every node in $C \setminus P$ is adjacent to a node on P. Equivalently, a tree is a caterpillar if its edges can be listed in such an order that every two consecutive edges in the list share an endpoint.

Throughout this section, we consider trees that span a given point set $\mathcal{P}$ in convex position. We say that (a drawing of) such a tree T is a *zigzagging caterpillar* if T is a caterpillar and the dual graph $T^\star$ of T is a path. Here, a *dual graph* $T^\star$ is defined as follows: Let C be a smooth closed curve passing through all points of $\mathcal{P}$. The curve bounds a convex region and the $n-1$ edges of T split that region into n subregions. The graph $T^\star$ has a node for each such subregion, and two nodes are connected if their subregions share an edge of T (see Figure 11).

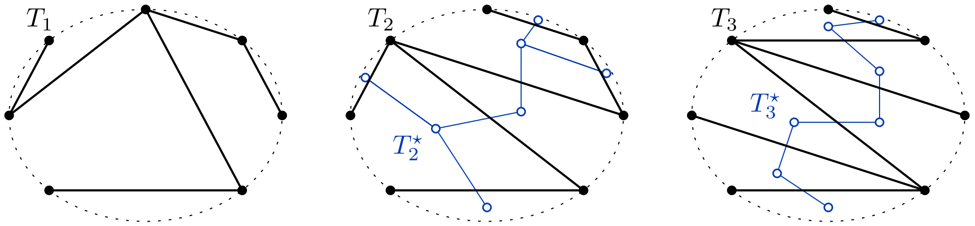

Fig. 11. Left: A tree T_1 that spans $\mathcal{P}$ but it is not a caterpillar. Middle: A tree T_2 that is a caterpillar but it is not zigzagging. Right: A tree T_3 that is a zigzagging caterpillar – the dual tree $T_3^\star$ is a path.

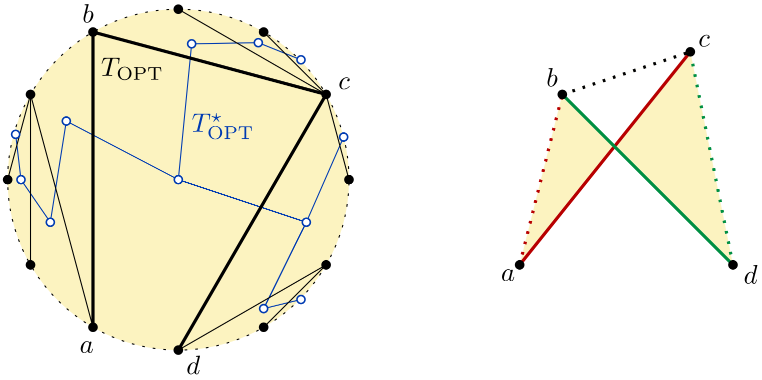

Fig. 12. Left: A point set in convex position with a (fake) longest plane tree T_{OPT} (black) and the dual graph $T^\star_{\text{OPT}}$ (blue). Right: If $T^\star_{\text{OPT}}$ has a node of degree three or more, a longer tree can be constructed by replacing one dotted edge of T_{OPT} by a longer edge (of the same color).

First, we prove that the longest plane tree of any set in convex position is a zigzagging caterpillar.

Theorem 1.2. *Let $\mathcal{P}$ be a finite point set in convex position in the plane. Then, every longest plane tree on $\mathcal{P}$ is a zigzagging caterpillar.*

Proof. Let T_{OPT} be a longest plane tree. We prove that $T^\star_{\text{OPT}}$ is a path. Suppose not, and consider a node in $T^\star_{\text{OPT}}$ of degree at least 3. Let ab, bc, cd be three corresponding edges of T_{OPT} (see Figure 12).

As $abcd$ is a convex quadrilateral, the triangle inequality gives $\|ab\| + \|cd\| < \|ac\| + \|bd\|$, so $\|ab\| < \|ac\|$ or $\|cd\| < \|bd\|$ (or both). Now, $T_1 = T_{\text{OPT}} \cup ac \setminus ab$ and $T_2 = T_{\text{OPT}} \cup bd \setminus cd$ are plane trees, and at least one of them is longer than T_{OPT}, a contradiction. □

Conversely, for each caterpillar C we will construct a point set $\mathcal{P}_C$ in convex position whose longest plane tree is isomorphic to C. In fact, $\mathcal{P}_C$ will be not only convex but also a *flat arc*. Formally, we say that a set $\mathcal{P}$ of $n = m + 1$ points $a_i = (x_i, y_i)$ satisfying $x_1 < \cdots < x_{m+1}$ is a *flat arc* if it is flat (that is, the absolute values of all y-coordinates are negligible) and the points $a_1, \dots, a_{m+1}$ all lie on the convex hull of $\mathcal{P}$ in this order.

We call the sequence $G(\mathcal{P}_C) = \{g_i\}_{i=1}^m = \{|x_{i+1} - x_i|\}_{i=1}^m$ the (horizontal) *gap sequence* of $\mathcal{P}_C$. Lastly, given a tree T spanning a flat arc $\mathcal{P}$, we define its *cover sequence* $\text{Cov}(T) = \{c_i\}_{i=1}^m$ to be a list storing the number of times each gap is "covered". Formally, c_i is the number of edges of T that contain a point whose orthogonal projection to the x-axis lies within the open interval (x_i, x_{i+1}), see Figure 13. Note that the gap sequence and the cover sequence determine the length of the

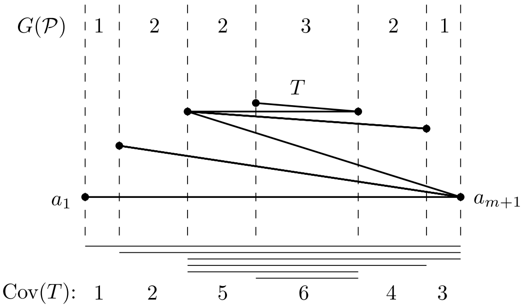

Fig. 13. Given a flat arc $\mathcal{P}$ of $n = m + 1$ points, we denote by $G(\mathcal{P}) = \{g_i\}_{i=1}^{m}$ its gap sequence. Given a tree T spanning $\mathcal{P}$, we denote by $\mathrm{Cov}(T) = \{c_i\}_{i=1}^{m}$ its cover sequence. Later we show that the longest plane tree T_{OPT} of a flat arc $\mathcal{P}$ is a zigzagging caterpillar containing the edge a_1a_{m+1} and that its cover sequence $\mathrm{Cov}(T_{\mathrm{OPT}})$ is a unimodal (single-peaked) permutation of $\{1, 2, \ldots, m\}$.

tree because $|T| = \sum_{i=1}^{m} c_i \cdot g_i$. Recall here that in a flat point set, the y-coordinates can be chosen arbitrarily small and thus we ignore them in our considerations.

Before we prove that any caterpillar can be the longest tree of a flat arc, we first show that the cover sequences of zigzagging caterpillars are unimodal (single-peaked) permutations.

Lemma 3.1. *Consider a flat arc $\mathcal{P}_C = \{a_1, \ldots, a_{m+1}\}$ and a zigzagging caterpillar T containing the edge a_1a_{m+1}. Then the cover sequence $\mathrm{Cov}(T)$ of T is a unimodal permutation of $\{1, 2, \ldots, m\}$.*

Proof. We show this lemma by induction on m. The case $m = 1$ is clear. Fix $m \geq 2$. By the definition of a zigzagging caterpillar, the dual graph $T^\star$ of T is a path. Since, by the assumption of the lemma, a_1a_{m+1} is an edge of T, either a_1a_m or a_2a_{m+1} is an edge of T too. Without loss of generality assume a_1a_m is an edge of T. Then $T \setminus \{a_1a_{m+1}\}$ is a zigzagging caterpillar on m points $a_1, \ldots, a_m$ containing the edge a_1a_m, hence by induction its cover sequence is a unimodal permutation of $\{1, 2, \ldots, m-1\}$. Adding the omitted edge a_1a_{m+1} adds 1 to each of the $m-1$ elements of the cover sequence and appends a 1 to the sequence, giving rise to a unimodal permutation of $\{1, 2, \ldots, m\}$. This completes the proof. □

Now we are ready to prove that any caterpillar, including a path, can be the longest plane tree of some flat arc.

Theorem 1.3. *For any caterpillar C, there exists a point set $\mathcal{P}$ in convex position such that the longest tree for $\mathcal{P}$ is unique and isomorphic to C.*

Proof. We will define a flat arc $\mathcal{P}_C$ such that the longest tree T_{OPT} of $\mathcal{P}_C$ is unique and isomorphic to C. Consider a flat arc $\mathcal{P} = \{a_1, \ldots, a_{m+1}\}$, with a yet unspecified gap sequence $\{g_i\}_{i=1}^{m}$. Take the caterpillar C and let T be its drawing onto $\mathcal{P}$ that contains the edge a_1a_{m+1} and is zigzagging (it is easy to see that such a drawing always exists). By Lemma 3.1 its cover sequence $\mathrm{Cov}(T) = \{c_i\}_{i=1}^{m}$ is a unimodal permutation of $\{1, 2, \ldots, m\}$. The total length $|T|$ of T can be expressed as $|T| = \sum_{i=1}^{m} c_i \cdot g_i$.

Now we specify the gap sequence: for $i = 1, \ldots, m$ set $g_i = c_i$. It is easy to see that this sequence defines a plane tree T; see Figure 13. It suffices to show that T constitutes the longest plane tree T_{OPT} of $\mathcal{P}$.

By Theorem 1.2, T_{OPT} is a zigzagging caterpillar. Also we prove by contradiction that, a_1a_{m+1} is an edge of T_{OPT}: suppose not. Since a_1a_{m+1} does not cross any other edge, adding it to T_{OPT} produces a plane graph with a single cycle C. All edges of T_{OPT} are shorter than a_1a_{m+1}, so omitting any other edge from C yields a longer plane tree, a contradiction.

We can thus apply Lemma 3.1 to see that $\mathrm{Cov}(T_{\text{OPT}})$ is a unimodal permutation π of $\{1, 2, \ldots, m\}$ and that $|T_{\text{OPT}}| = \sum_{i=1}^{m} \pi_i \cdot g_i$. As c_i and g_i match and as c, g, and π are permutations, the Cauchy-Schwarz inequality gives

$$|T_{\text{OPT}}| = \sum_{i=1}^{m} \pi_i \cdot g_i \leq \sqrt{\sum_{i=1}^{m} \pi_i^2 \cdot \sum_{i=1}^{m} g_i^2} = \sum_{i=1}^{m} c_i^2 = |T|, \text{ with equality iff } \pi_i = c_i, \text{ for all } i. \tag{23}$$

Therefore, T_{OPT} is unique and $T_{\text{OPT}} = T$ as desired. □

3.2 Upper Bounds on BoundDiam (d)

The algorithms for approximating $|T_{\text{OPT}}|$ often produce trees with small diameter. Given an integer $d \geq 2$ and a point set $\mathcal{P}$, let $T_{\text{OPT}}^{d}(\mathcal{P})$ be a longest plane tree spanning $\mathcal{P}$ among those whose diameter is at most d. One can then ask what is the approximation ratio

$$\mathrm{BoundDiam}(d) = \inf_{\mathcal{P}} \frac{|T_{\text{OPT}}^{d}(\mathcal{P})|}{|T_{\text{OPT}}(\mathcal{P})|}$$

achieved by all such trees. As before, we drop the dependency on $\mathcal{P}$ in the notation and just use T_{OPT}^{d} and T_{OPT}.

When $d = 2$, this reduces to asking about the performance of stars. A result due to Alon et al. [1, Theorem 4.1] can be restated as $\mathrm{BoundDiam}(2) = 1/2$. Below we show a crude upper bound on $\mathrm{BoundDiam}(d)$ for general d and then a specific upper bound tailored to the case $d = 3$. Note that Theorem 1.6 shows that $|T_{\text{OPT}}^{3}|$ can be computed in polynomial time. Our proofs in this section use the notions of flat arc, gap sequence and cover sequence defined in section 3.1.

Theorem 1.4. *For every $d \geq 2$, there exists a point set $\mathcal{P}$ in convex position so that every plane tree of diameter at most d on $\mathcal{P}$ is at most*

$$1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3)$$

times as long as the length $|T_{\text{OPT}}|$ of a longest (general) plane tree on $\mathcal{P}$. It follows that

$$\mathrm{BoundDiam}(d) \leq 1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3).$$

Proof. Let T be a path with $d + 1$ edges (and diameter $d + 1$). Then, T is a zigzagging caterpillar whose cover sequence is $G = (1, 3, 5, \ldots, d + 1, \ldots, 6, 4, 2)$ By Theorem 1.3, there exists a point set $\mathcal{P}$ for which T is isomorphic to the unique longest spanning tree of $\mathcal{P}$. In fact, the proof of Theorem 1.3 gives an explicit construction for $\mathcal{P}$: it is a flat arc on $d + 2$ points with gap sequence G. Hence, we can assume that all the gaps in $\mathcal{P}$ are integers.

Therefore, any other plane spanning tree $T' \neq T$ on $\mathcal{P}$ must have a length that is an integer less than $|T|$. Using that $|T| = \sum_{i=1}^{d+1} i^2 = \frac{1}{6}(d+1)(d+2)(2d+3) = \frac{1}{3}d^3 + o(d^3)$ we obtain

$$\mathrm{BoundDiam}(d) \leq \frac{|T| - 1}{|T|} = 1 - \frac{6}{(d+1)(d+2)(2d+3)}.$$ □

For $d = 3$, Theorem 1.4 gives $\mathrm{BoundDiam}(3) \leq 29/30$. By tailoring the point set size, the gap sequence $\{g_i\}_{i=1}^{m}$, and by considering flat point sets in convex position that are not arcs we improve the bound to $\mathrm{BoundDiam}(3) \leq 5/6$.

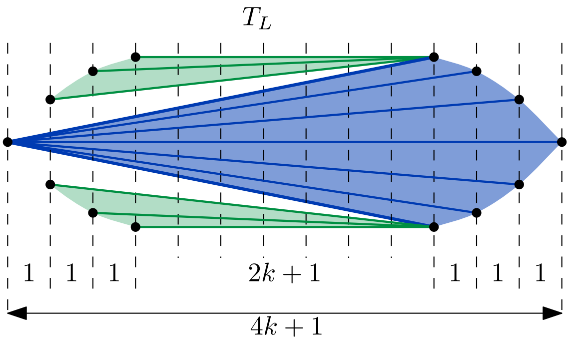

Fig. 14. An illustration of the point set $\mathcal{P}_{4k+2}$ when $k = 3$ with a tree T_L (blue and green).

Theorem 1.5. *For every $\varepsilon > 0$, there exists a point set $\mathcal{P}$ in convex position such that every longest plane tree of diameter three on $\mathcal{P}$ is at most $(5/6) + \varepsilon$ times as long as every longest (general) plane tree on $\mathcal{P}$.*

Proof. Consider a flat point set $\mathcal{P}_{4k+2}$ consisting of two flat arcs that are symmetric with respect to a horizontal line ℓ', each with a gap sequence

$$(\underbrace{1, \ldots, 1}_{k\times},\ 2k + 1,\ \underbrace{1, \ldots, 1}_{k\times}).$$

In other words, $\mathcal{P}_{4k+2}$ consists of two diametrically opposite points, four unit-spaced arcs of k points each, and a large gap of length $2k + 1$ in the middle (see Figure 14).

On one hand, straightforward counting gives $|T_{\text{OPT}}| \geq |T_L| = 12k^2 + 6k + 1$, where T_L is the tree depicted in Figure 14. On the other hand, we claim that any tree T with diameter at most three has length at most $10k^2 + 6k + 1$. Thus

$$\text{BoundDiam}(3) \leq \frac{10k^2 + 6k + 1}{12k^2 + 6k + 1},$$

which tends to $5/6$ as $k \longrightarrow \infty$.

In the rest of this proof we will show that the longest tree T among those with diameter at most 3 on $\mathcal{P}_{4k+2}$ has length at most $10k^2 + 6k + 1$. First, note that as T has diameter at most 3 it is either a star or it has a cut edge ab whose removal decomposes T into a star centered at a and a star centered at b.

To add the lengths of the edges of the tree, we will often split the edges across the large gap into two parts: one part contained within the left or the right part and with maximum length k, and one part going across the large gap, with a length between $2k + 1$ and $2k + 1 + k = 3k + 1$.

If T is a star, then without loss of generality we can assume that its root a is to the left of the large gap. Denote the distance from the large gap to a by i (we have $0 \leq i \leq k$). A straightforward calculation (see Figure 15) then gives

$$\begin{aligned}|S_a| &= (k - i)^2 + i(i + 1) + (2k + 1) \cdot (i + 2k + 1) + k^2 \\ &\leq k(k + 1) + (2k + 1)(3k + 1) + k^2 \\ &= 8k^2 + 6k + 1.\end{aligned}$$

Now suppose T has diameter 3 and denote its cut edge by ab. If ab is vertical then $|T| = |S_a|$ and the above bound applies. In the remaining cases, without loss of generality, we can assume that a

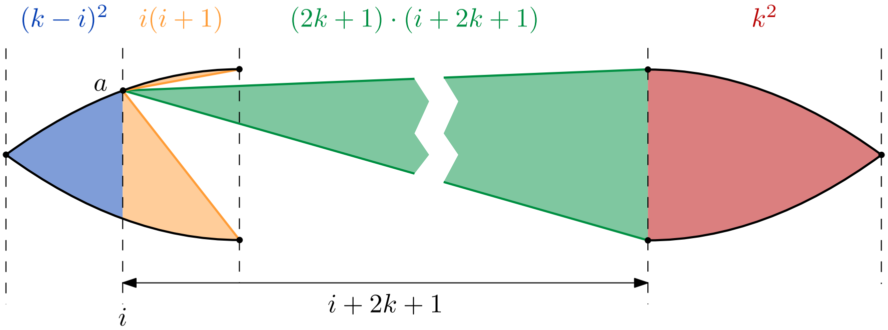

Fig. 15. The longest star for $\mathcal{P}_{4k+2}$ is short.

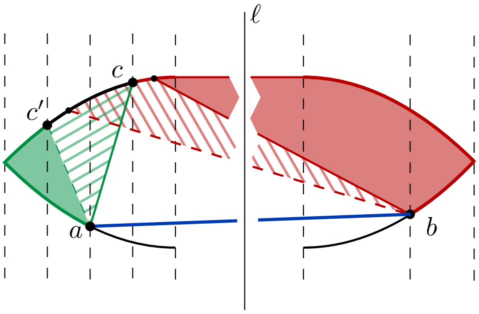

Fig. 16. Points on one side of ab are either all connected to a or all to b.

is to the left of b. The line ab then splits the remaining points of $\mathcal{P}_{4k+2}$ into two arcs – one above ab and the other one below it.

We claim that in the longest tree with diameter 3, the points of one arc are either all connected to a or all to b: for the sake of contradiction, fix an arc from a to b. Without loss of generality, take the clockwise arc from a to b. Suppose c is the last point (in this order) connected to a. See Figure 16. Then by the choice of c, all points after c are connected to b and by convexity, all points before c are connected to a. Let ℓ be the vertical line through the midpoint of ab. If c lies to the left of ℓ, replacing the edge ac by bc increases the length of the tree, while keeping the tree plane. Otherwise, connecting the clockwise neighbor of c to a and removing its connection to b, increases the length of the tree and keeps it plane. Thus we either have $c = a$ or $c = b$ as claimed.

Since $\mathcal{P}_{4k+2}$ is centrally symmetric and we have already dealt with the case when T is a star, we can without loss of generality assume that all points above ab are connected to a and all points below ab are connected to b. Now suppose that a is below the diameter line ℓ' of $\mathcal{P}_{4k+2}$ and consider the reflection $a' \in \mathcal{P}_{4k+2}$ of a about ℓ'. Then the tree T' with cut edge $a'b$ is longer than T, since in T' the points to the left of aa' are connected to b rather than to a and all the other edges have the same length (see Figure 17). Hence we can assume that a lies above ℓ' (or on it). Similarly, b lies below ℓ' or on it.

It remains to distinguish two cases based on whether a and b lie on different sides of the large gap or not. Either way, denote the distance from the gap to a and b by i and j, respectively. In the first case, considering the subdivision of the edges as sketched in Figure 18, a straightforward

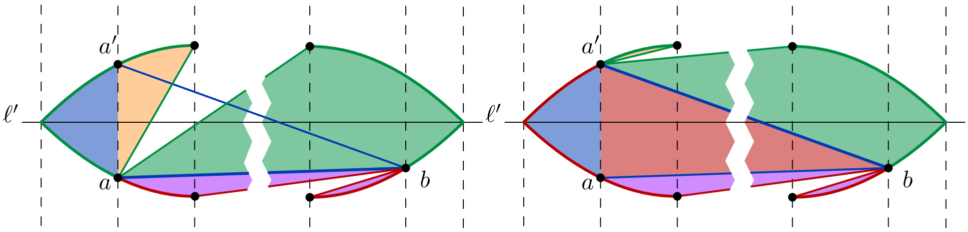

Fig. 17. Flipping a to a' increases the total length.

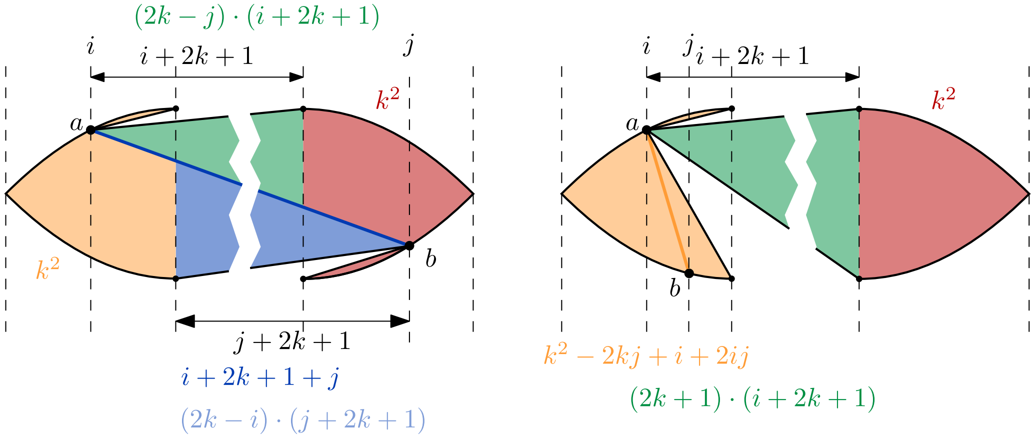

Fig. 18. Decomposition of the edges for the upper bounds, case 1 is seen left, case 2 is to the right.

calculation gives:

$$\begin{aligned}|T| &= k^2 + (2k-j)(i+2k+1) + (i+2k+1+j) + (2k-i)(2k+1+j) + k^2\\ &= 10k^2 + 6k + 1 - 2ij \leq 10k^2 + 6k + 1\end{aligned}$$

with equality if and only if either a or b (or both) lie on the boundary of the large gap. In the second case, since a is to the left of b we have $i > j$ and a similar calculation, see Figure 18, gives

$$|T| = (k^2 - 2kj + i + 2ij) \ + \ (2k+1)(2k+1+i) \ + \ k^2.$$

Since the right-hand side is increasing in i and $i \leq k$, we get

$$|T| \leq (k^2 + k) + (2k+1)(3k+1) + k^2 = 8k^2 + 6k + 1$$

which is less than the claimed upper bound $10k^2 + 6k + 1$. □

4 Polynomial Time Algorithms for Small Diameters

In Section 4.1 we show that the longest plane tree of diameter at most three can be computed in polynomial time using dynamic programming. Our approach bears some resemblance to the polynomial time plane matching algorithm of Aloupis et al. [2]. The main challenge in our case is the efficient implementation of the dynamic programming algorithm. Such a tree may be relevant in providing an approximation algorithm with a better approximation factor.

In Section 4.2 we show how to compute in polynomial time a longest plane tree among those of the following form: all the points are connected to three distinguished points on the boundary of the convex hull. Again, such a tree may play an important role in designing future approximation

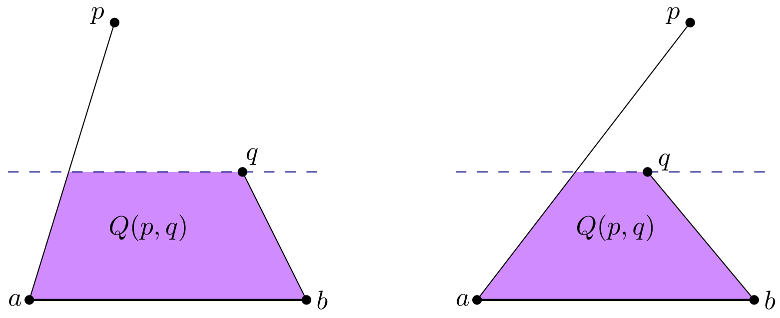

Fig. 19. Two examples of valid pairs p, q with their quadrilaterals $Q(p, q)$ shaded.

algorithms, as intuitively it seems to be better than the three stars considered in our previous approximation algorithm, when there is a point in the far region.

4.1 Finding the Longest Tree of Diameter 3

For any two points a, b of $\mathcal{P}$, a *bistar rooted* at a and b is a tree that contains the edge ab and where each point in $\mathcal{P} \setminus \{a, b\}$ is connected to either a or b. Note that every star is also a bistar. In this section we prove the following theorem.

THEOREM 1.6. *For any set $\mathcal{P}$ of n points in general position, a longest plane tree of diameter at most three on $\mathcal{P}$ can be computed in $O(n^4)$ time.*

PROOF. Note that each bistar has diameter at most 3. Conversely, each tree with diameter three has one edge uv where each point $p \in \mathcal{P}$ has distance at most 1 to either u or v. It follows that all trees with diameter at most three are bistars. In Lemma 4.2 we show that for any two fixed points a, b, the longest plane bistar rooted at a and b can be computed in time $O(n^2)$. By iterating over all possible pairs of roots and taking the longest such plane bistar, we find the longest plane spanning tree with diameter at most 3 in time $O(n^4)$. □

We now describe the algorithm used to find the longest plane bistar rooted at two fixed points a and b. Without loss of generality, we can assume that the points a and b lie on a horizontal line with a to the left of b. Furthermore, as we can compute the best plane bistar above and below the line through a and b independently, we can assume that all other points lie above this horizontal line. In order to solve this problem by dynamic programming, we consider a suitable subproblem.

The subproblems considered in the dynamic programming algorithm are indexed by an ordered pair p, q of different points of $\mathcal{P}$ such that the edges ap and bq do not cross. A pair p, q satisfying these condition is a *valid pair*. For each valid pair p, q, note that the segments ap, pq, qb and ba form a simple (possibly non-convex) quadrilateral. Let $Q(p, q)$ be the (convex) portion of this quadrilateral below the horizontal line $y = \min\{y(p), y(q)\}$, as shown in Figure 19. We define the value $Z(p, q)$ to be the length of the longest plane bistar rooted at a and b on the points in the interior of $Q(p, q)$, without counting $\|ab\|$. If there are no points of $\mathcal{P}$ within the quadrilateral $Q(p, q)$, we set $Z(p, q) = 0$.

If the quadrilateral $Q(p, q)$ contains some points from $\mathcal{P}$, we let $k_{p,q}$ be the highest point of $\mathcal{P}$ inside of $Q(p, q)$. Then we might connect $k_{p,q}$ either to a or to b. By connecting it to a, we force all points in the triangle $L_{p,q}$ defined by the edges ap and $ak_{p,q}$ and the line $y = y(k_{p,q})$, to be connected to a. Similarly, when connecting $k_{p,q}$ to b, we enforce the triangle $R_{p,q}$ defined by bq, $bk_{p,q}$ and the line $y = y(k_{p,q})$. See Figure 20 for an illustration. In the former case we are left with the subproblem defined by the valid pair $k_{p,q}, q$, while in the latter case we are left with the subproblem defined by

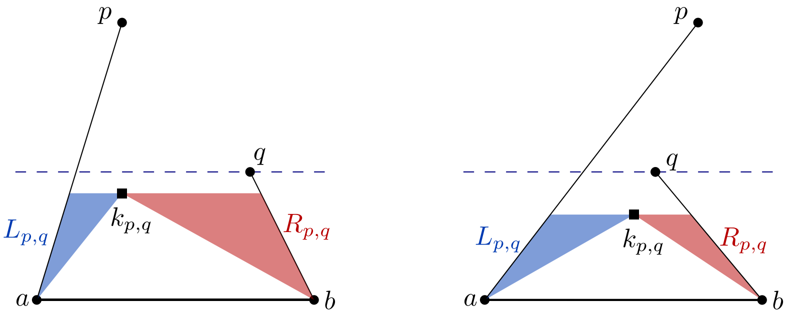

Fig. 20. The highest point $k_{p,k}$ defines two triangular regions and in one of them the edges are forced.

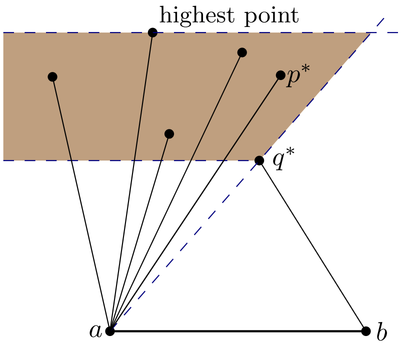

Fig. 21. The point p^* is the point of $\mathcal{P}$ in the shaded region that is angularly around a closest to q^*.

the valid pair $p, k_{p,q}$. Formally, for each valid pair p, q we have the following recurrence:

$$Z(p,q) = \begin{cases} 0 & \text{if no point of } \mathcal{P} \text{ is in } Q(p,q), \\ \max \begin{cases} Z(k_{p,q}, q) + \|ak_{p,q}\| + \sum_{l \in L_{p,q}} \|al\| \\ Z(p, k_{p,q}) + \|bk_{p,q}\| + \sum_{r \in R_{p,q}} \|br\| \end{cases} & \text{otherwise.} \end{cases}$$

Lemma 4.1. *Using $Z(p,q)$ for all valid p, q we can find a longest plane bistar rooted at a and b.*

Proof. Consider a fixed longest plane bistar and assume, without loss of generality, that the highest point is connected to a; the other case is symmetric. Suppose that there is at least one point that is connected to b (otherwise, the bistar degenerates to a star centered at a), and let q^* be the highest point that is connected to b. This means that all the points above q^* are attached to a. Let p^* be the point above q^* that, circularly around a, is closest to ab. See Figure 21. Note that p^*, q^* is a valid pair. For $p \in \mathcal{P}$, denote by L_p the set of points in $\mathcal{P}$ that, circularly around a, are to the left of ap. Similarly, denote by R_q the set of points in $\mathcal{P}$ below q and to the right of bq, when sorted circularly around b, see Figure 22. The length of this optimal plane bistar rooted at a and b is then

$$\left(\sum_{l \in L_{p^*}} \|al\|\right) + \left(\sum_{r \in R_{q^*}} \|br\|\right) + \|ap^*\| + \|bq^*\| + \|ab\| + Z(p^*, q^*).$$

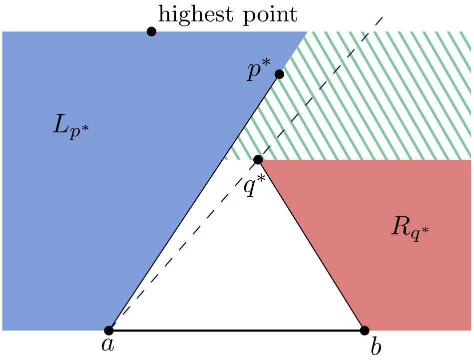

Fig. 22. The regions L_{p^*} and R_{q^*} are shaded. The region with bars is empty.

On the other hand, each of the values of the form

$$\left(\sum_{l \in L_p} \|al\|\right) + \left(\sum_{r \in R_q} \|br\|\right) + \|ap\| + \|bq\| + \|ab\| + Z(p,q), \tag{24}$$

where p, q is a valid pair of points such that $y(p) > y(q)$, is the length of a plane, not necessarily spanning, bistar rooted at a and b. (It is not spanning, if there is some point above q and right of ap.)

Taking the maximum of $|S_a|$ and expression (24) over the valid pairs p, q such that $y(p) > y(q)$ gives the longest plane bistar for which the highest point is connected to a. A symmetric formula gives the best plane bistar if the highest point is connected to b. Taking the maximum of both cases yields the optimal value.

The actual edges of the solution can be backtracked by standard methods. □

Lemma 4.2. *The algorithm described in the Proof of Lemma 4.1 can be implemented in time $O(n^2)$.*

Proof. A main complication in implementing the dynamic programming algorithm and evaluating (24) efficiently is finding sums of the form $\sum_{l \in \Delta \cap \mathcal{P}} \|al\|$ or $\sum_{r \in \Delta \cap \mathcal{P}} \|br\|$, and the highest point in $\Delta \cap \mathcal{P}$, where Δ is a query triangle (with one vertex at a or b). These type of range searching queries can be handled using standard data structures [14, 24]: after preprocessing $\mathcal{P}$ in time $O(n^2 \operatorname{polylog} n)$, any such query can be answered in $O(\operatorname{polylog} n)$ time. Noting that there are $O(n^2)$ such queries, a running time of $O(n^2 \operatorname{polylog} n)$ is immediate. However, exploiting our specific structure and using careful bookkeeping we can get the running time down to $O(n^2)$.

As a preprocessing step, we first compute two sorted lists $\mathcal{L}_a$ and $\mathcal{L}_b$ of $\mathcal{P} \setminus \{a, b\}$. The list $\mathcal{L}_a$ is sorted by the circular ordering around a and $\mathcal{L}_b$ by the circular ordering at b.

The values $\sum_{l \in L_p} \|al\|$ and $\sum_{r \in R_q} \|br\|$ (as depicted in Figure 22) for all $p, q \in \mathcal{P}$ can be trivially computed in $O(n^2)$ time: there are $O(n)$ such values and for each of them we can scan the whole point set to explicitly get L_p or R_q in $O(n)$ time. There are faster ways of doing this, but for our result this suffices.

Assuming that the values $Z(p,q)$ are already available for all valid pairs p, q, we can evaluate (24) in constant time. For any two points p, q we check whether they form a valid pair and whether $y(p) > y(q)$ in constant time. We can then either evaluate (24) or the symmetric formula again in constant time. Thus, in $O(n^2)$ time we obtain the optimal solution.

It remains to compute the values $Z(p,q)$ for all valid pairs p, q. First, we explain how to compute all the triples of the form $(p, q, k_{p,q})$ for all valid pairs p, q in $O(n^2)$ time. Then we group these triples in a clever way to implement the dynamic programming algorithm efficiently.

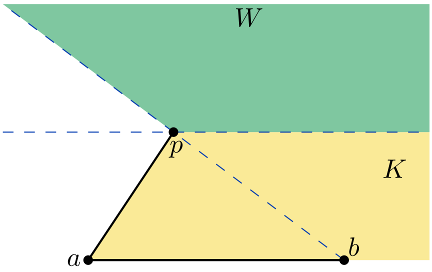

Fig. 23. Left: The regions defining W and K for fixed a, b and p.

We focus on the triples with $y(p) < y(q)$ and show how to find the triples of the form $(p, \cdot, \cdot)$ for a fixed p. The case with $y(p) > y(q)$ and a fixed q is symmetric. Let W be the points of $\mathcal{P}$ to the right of the ray bp above the horizontal line $y = y(p)$. Furthermore let K be the points of $\mathcal{P}$ to the right of ap and with y coordinate between $y(a)$ and $y(p)$. An illustration of W and K can be found in Figure 23. Any point q with $y(q) > y(p)$ forms a valid pair p, q if and only if q lies in W. The point $k_{p,q}$ must lie in K by its definition.

We use $\mathcal{L}_b$ to find the triples $(p, \cdot, \cdot)$. We iterate through the list in clockwise order starting at the ray ba and keep track of the highest $k^* \in K$ encountered so far. If the current point lies in $\mathcal{P} \setminus (K \cup W)$ we simply skip it. If the current point lies in K we update k^* if necessary. Finally, if the current point lies in W, we report the triple $(p, q, k^* = k_{p,q})$ with q being the current point.

For a fixed p we only iterate $\mathcal{L}_b$ once. Thus, for this fixed p the running time for finding all triples $(p, q, k_{p,q})$ with p, q forming a valid pair and $y(p) < y(q)$ is $O(n)$. By the procedure we just described and its symmetric procedure on all $p \in \mathcal{P} \setminus \{a, b\}$ we find all the triples $(p, q, k_{p,q})$ where p, q is a valid pair in overall $O(n^2)$ time.

To compute $Z(p, q)$ for all valid pairs using dynamic programing, we also need to compute the corresponding values $\sum_{l \in L_{p,q}} \|al\|$ and $\sum_{r \in R_{p,q}} \|br\|$. Refer to Figure 20 to recall the definition of $L_{p,q}$ and $R_{p,q}$. For the following procedure we shift the focus to the point k. For each point $k \in \mathcal{P} \setminus \{a, b\}$ we collect all triples $(p, q, k = k_{p,q})$, and compute the sums $\sum_{l \in L_{p,q}} \|al\|$ and $\sum_{r \in R_{p,q}} \|br\|$ in linear time for each fixed k, as follows.

We concentrate on the first type of sum, $\sum_{l \in L_{p,q}} \|al\|$, where q has no role, as the sum is defined by p and k. For the following description we assume that $\mathcal{L}_a$ is sorted in counterclockwise order. We create a (sorted) subsequence $\mathcal{L}_a^k$ of $\mathcal{L}_a$ containing only the points with y coordinate below $y(k)$ and an angle at a larger than *anglebak*. We iterate over the elements of $\mathcal{L}_a$, starting at the successor of k. While iterating we maintain the last element from $\mathcal{L}_a^k$ we have seen and the prefix sum $\sum_l \|al\|$ of all points l in $\mathcal{L}_a^k$ we encountered so far. When advancing to the next point p in $\mathcal{L}_a$ there are two possibilities. If p also lies in $\mathcal{L}_a^k$ we advance in $\mathcal{L}_a^k$ and update the prefix sum. Otherwise, we can report $\sum_{l \in L_{p,q'}} \|al\|$ to be the current prefix sum for all q' in a triple (p, q', k).

For any fixed k this needs at most two iterations through the list $\mathcal{L}_a$ and can thus be executed in $O(n)$ time. A symmetric procedure can be carried out for $\sum_{r \in R_{p,q}} \|br\|$ using $\mathcal{L}_b$. As in the case for finding the triples, this results in $O(n^2)$ overall time to compute all relevant sums $\sum_{l \in L_{p,q}} \|al\|$ and $\sum_{r \in R_{p,q}} \|br\|$.

Now we can implement the dynamic programming algorithm in the straightforward way. Using the precomputed information we spend $O(1)$ time for each value $Z(p, q)$, for a total running time of $O(n^2)$. □

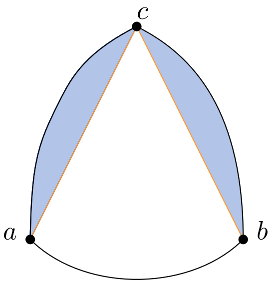

Fig. 24. The regions cut off by ac and bc can be solved independently as bistars.

4.2 Extending the Approach to Special Trees of Diameter 4

Now we show how we can extend the ideas to get a polynomial time algorithm for special trees with diameter 4. Given three points a, b, c of $\mathcal{P}$, a *tristar rooted* at a,b, and c is a tree such that every $v \in \mathcal{P} \setminus \{a, b, c\}$ is connected to exactly one of a, b, or c. We show the following:

Theorem 1.7. *For any set $\mathcal{P}$ of points in general position and for any three specified points on the boundary of the convex hull of $\mathcal{P}$, we can compute in polynomial time the longest plane tree such that each edge is incident to at least one of the three specified points.*

Proof. Let a, b, c be the specified points on the boundary of the convex hull of $\mathcal{P}$. We have to compute the longest plane tristar rooted at a, b and c. We assume, without loss of generality, that the edges ac and bc are present in the tristar; the other cases are symmetric. We further assume that a and b lie on a horizontal line, with a to the left of b.

The subproblem defined by the regions to the left of ac and to the right of bc, as depicted in Figure 24, can be solved independently of the rest of the instance. Indeed, the presence of the edges ac and bc blocks any edge connecting a point in one of those regions to a point outside the region. Each one of the subproblems defined by these regions can be solved as plane bistars, one rooted at a, c and one rooted at b, c. It remains to solve the problem for the points enclosed by ac, cb and the portion of the boundary of the convex hull from b to a (clockwise). Let Q be this region. The problem for Q can also be solved independently of the other problems. We assume for the rest of the argument that all points of $\mathcal{P}$ are in Q.

To solve the problem for Q we will use a variation of the dynamic programming approach for bistars. For any two points p, q of $\mathcal{P}$, let us write $p \preceq_c q$ when in the counterclockwise order around c we have the horizontal ray through c to the left, then cp and then cq or the segments cp and cq are collinear. (Since we assume general position, the latter case occurs when $p = q$.)

The subproblems for the dynamic programming algorithm are defined by a 5-tuple (p, p', r, q', q) of points of $\mathcal{P}$ such that

- p' and q' are distinct;
- $p \preceq_c p' \preceq_c r \preceq_c q' \preceq_c q$; and
- p' and q' are contained in the closed triangle cpq.

One such five tuple is a *valid tuple*, see Figure 25. Note that the first and the second condition imply that ap and bq do not cross. Some of the points may be equal in the tuple; for example we may have $p = p'$ or $p' = r$ or even $p = p' = r$. For each valid tuple (p, p', r, q', q), let $Q(p, r, q)$ be the points of $\mathcal{P}$ contained in Q below the polygonal path ap, pq and qb, and below the horizontal line through the lowest of the points p, q, r. Note that the point r is used only to define the horizontal line.

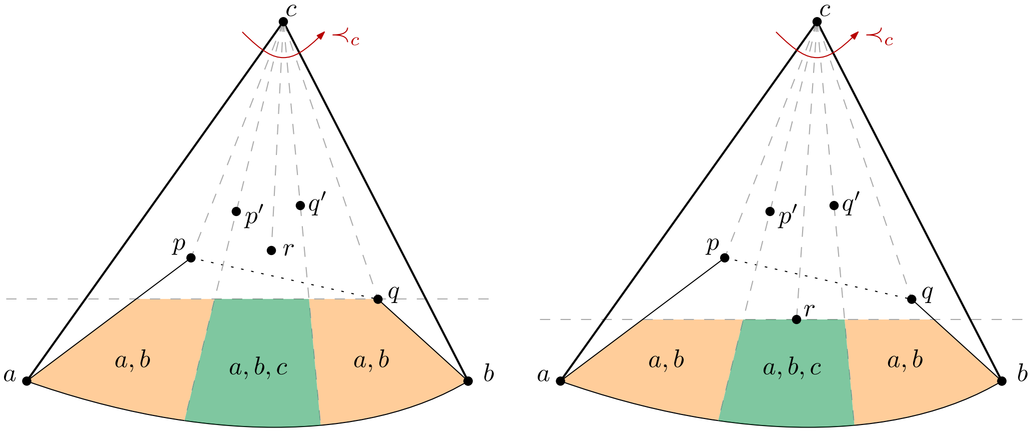

Fig. 25. Two examples of valid tuples (p, p', r, q', q) with the region $Q(p, r, q)$ shaded. Each $Q(p, r, q)$ is split into 3 regions telling, for each of them, which roots can be used for that region.

For each valid tuple (p, p', r, q', q), we define $Z(p, p', r, q', q)$ as the length of the optimal plane tristar rooted at a, b, c for the points in the interior of $Q(p, r, q)$, without counting $\|ac\| + \|bc\|$, and with the additional restriction that a point k can be connected to c only if $p' \preceq_c k \preceq_c q'$. This last condition is equivalent to telling that no edge incident to c in the tristar can cross ap' or bq'. Thus, we are looking at the length of the longest plane forest in which each point in the interior of $Q(p, r, q)$ must be connected to either a, b or c, and no edge crosses ap' or bq'.

If $Q(p, r, q)$ contains no points, then $Z(p, p', r, q', q) = 0$. If $Q(p, r, q)$ contains some point, we find the highest point $k = k(p, r, q)$ in $Q(p, r, q)$ and consider how it may attach to the roots and which edges are enforced by each of these choices. We can always connect k to a or b, as $Q(p, r, q)$ is free of obstacles. If $p' \preceq_c k \preceq_c q'$, then we can connect k to c. If k is connected to c, the edge kc does not obstruct any other edges in $Q(p, r, q)$. Thus, connecting k to c only lowers the horizontal line that bounds the region $Q(\cdot, \cdot, \cdot)$ that has to be considered. However if k is connected to a or b, it splits off independent regions, some of them can be attached to only one of the roots, some of them are essentially a bistar problem rooted at c and either a or b. To state the recursive formulas precisely, for a region R and roots z, c, let $\mathrm{BS}_{z,c}(R)$ be the length of the optimal plane bistar rooted at z, c for the points in R. Such a value can be computed using Lemma 4.1.

Formally the recurrence for $Z(p, p', r, q', q)$ looks as follows. If $Q(p, r, q)$ is empty, then we have $Z(p, p', r, q', q) = 0$. If $Q(p, r, q)$ is not empty, let k be the highest point of $Q(p, r, q)$; we have three subcases:

Case 1: $k \prec_c p'$. Let L be the set of points of $Q(p, r, q)$ above ak. Let B be the set of points of $Q(p, r, q)$ below bk and let G be the set of points t of $Q(p, r, q) \setminus B$ such that $k \prec_c t \prec_c p'$. Let g be the point with largest angle at b among the points in G (or $g = k$ if G is empty. Then we define R' to be the set of all points t in $Q(p, r, q) \setminus B$ with $p' \prec_c t \prec_c q'$ which are above gb. Finally let $R = Q(p, q, r) \setminus (B \cup R')$, as shown in Figure 26. We get the following recurrence in this case:

$$Z(p, p', r, q', q) = \max \begin{cases} Z(k, p', r, q', q) + \|ak\| + \sum_{l \in L} \|al\| \\ \mathrm{BS}_{a,b}(B) + \|bk\| + \mathrm{BS}_{b,c}(R') + \sum_{r \in R} \|br\|. \end{cases}$$

The recurrence comes from the two possibilities of how we can connect k: if we connect k to a, then we are forced to connect all points in L to a, and we can recurse on the subproblem where k takes the role of a. If we connect k to b, then the points in the region B can connect

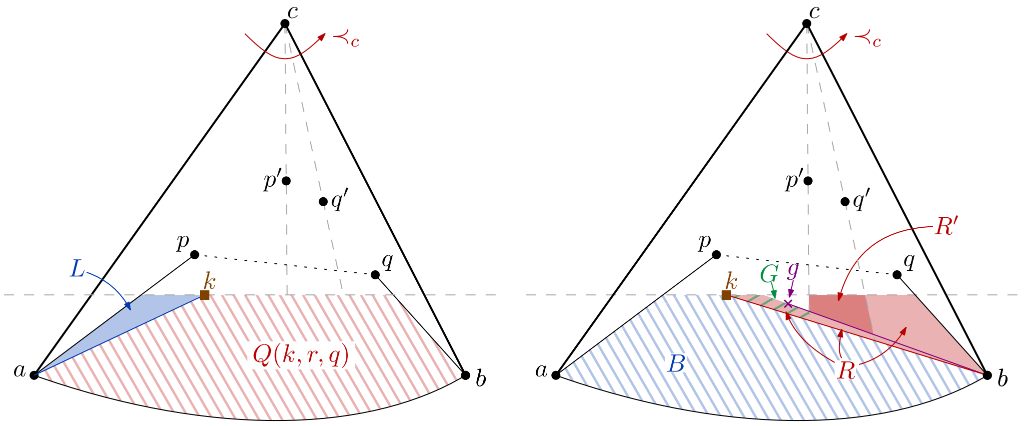

Fig. 26. The two cases in the recurrence when $k \prec_c p'$.

only to a and b, the points in the region R can connect only to b, and the points in the region R' can connect to b and c.

Case 2: $p' \prec_c k \prec_c q'$. Let L' be the set of points t of $Q(p,r,q)$ above ak such that $t \prec_c p'$ and let L be the set of points t of $Q(p,r,q)$ above ak such that $p' \prec_c t \prec_c k$. Furthermore, let R be the set of points t of $Q(p,r,q)$ above bk such that $k \prec_c t \prec_c q'$, and let R' be the set of points t of $Q(p,r,q)$ above bk such that $q' \prec_c t$; see Figure 27. We get the following recurrence in this case:

$$Z(p,p',r,q',q) = \max \begin{cases} Z(p,p',k,q',q) + \|ck\| \\ Z(k,k,k,q',q) + \|ak\| + \mathrm{BS}_{a,c}(L) + \sum_{l \in L'} \|al\| \\ Z(p,p',k,k,k) + \|bk\| + \mathrm{BS}_{b,c}(R) + \sum_{r \in R'} \|br\| \end{cases}$$

Again, the recurrence comes from the two possibilities of how we can connect k: if we connect k to c, then we recurse on the subproblem where k takes the role of r. If we connect k to a, then all the points in the region L' can connect only to a, all the points in the region L can connect to a or c, and we recurse on the subproblem where k takes the role of p, p', and r. Symmetrically, if we connect k to b, then all the points in the region R' can connect only to b, all the points in the region R can connect to b or c, and we recurse on the subproblem where k takes the role of q, q', and r.

Case 3: $q' \prec_c k$. The case is symmetric to the case $k \prec_c p'$.

The values $Z(p,p',r,q',q)$ for all valid tuples (p,p',r,q',q) can be computed using dynamic programming and the formulas described above. The dynamic programming algorithm is correct by the correctness of the recurrence relation.

Note that we have to compute a few solutions for bistars. In the recursive calls these have the form $\mathrm{BS}_{z,c}(R)$ for $z \in \{a,b\}$ and some region R of constant size description. More precisely, each such region is defined by two points of $\mathcal{P}$ (case of L in Figure 27) or by four points of $\mathcal{P}$ (case of R' in Figure 26, where it is defined by k,g,q',q). Thus, we have $O(n^4)$ different bistar problems, and each of them can be solved in polynomial time using Lemma 4.1.

To compute the optimal plane tristar, we add three dummy points to $\mathcal{P}$ before we start the dynamic programming. We add a point c_a on the edge ac arbitrarily close to c, a point c_b on the edge bc arbitrarily close to c, and a point c' between c_a and c_b, see Figure 28. We perturb the points c_a, c' and c_b slightly to get them into general position. Note that (c_a, c_a, c', c_b, c_b) is a

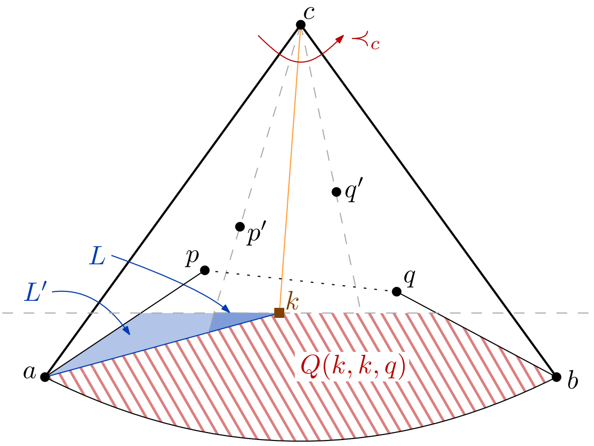

Fig. 27. One of the cases in the recurrence when $p' \prec_c k \prec_c q'$.

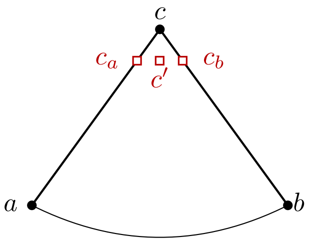

Fig. 28. Starting case.

valid tuple. We can now compute $Z(c_a, c_a, c', c_b, c_b)$ by implementing the recurrence with standard dynamic programming techniques in polynomial time. By adding $\|ac\|$, $\|bc\|$ and the lengths of the independent bistars from Figure 24 to $Z(c_a, c_a, c', c_b, c_b)$, we get the length of the longest plane tristar rooted at a, b and c. □

Similar to the proof Theorem 1.6 we can iterate over all possible triples of specified points to find the best plane tristar rooted at the boundary of the convex hull.

5 Local Improvements Fail

One could hope that the longest plane spanning tree problem could perhaps be solved by either a greedy approach or by a local search approach [37]. It is easy to find point sets on as few as 5 points where the obvious greedy algorithm fails to find the longest plane tree. In this section, we show that the following natural local search algorithm `AlgLocal(`$\mathcal{P}$`)` fails too:

```
Algorithm AlgLocal($\mathcal{P}$):
(1) Construct an arbitrary plane spanning tree $T$ on $\mathcal{P}$.
(2) While there exists a pair of points $a, b$ such that $T \cup \{ab\}$ contains an edge $cd$ with $\|cd\| < \|ab\|$ and $T \cup \{ab\} \setminus \{cd\}$ is a plane tree:
    (a) Set $T \to T \cup \{ab\} \setminus \{cd\}$.    // tree $T \cup \{ab\} \setminus \{cd\}$ is longer than $T$
(3) Output $T$.
```

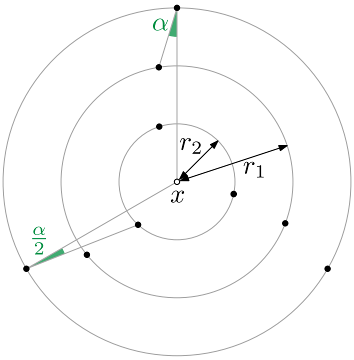

Fig. 29. Construction of the point set.

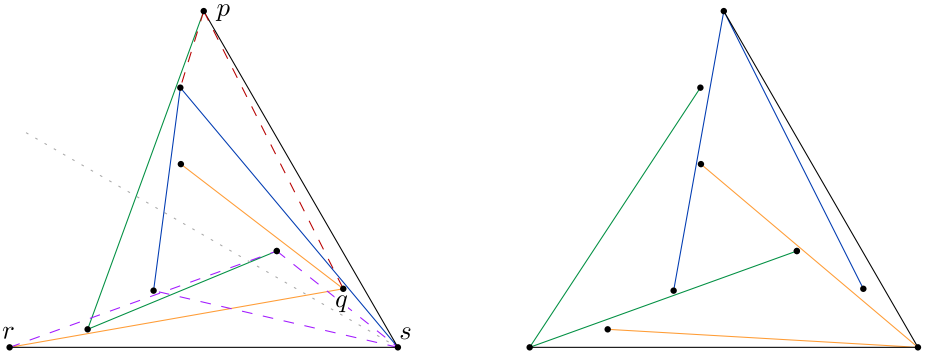

Fig. 30. Left: A tree which cannot be locally improved; Right: A tree where each pair of edges in the same color is at least as long as the matching pair to the left.

Lemma 5.1. *There are point sets $\mathcal{P}$ for which the algorithm* `AlgLocal`$(\mathcal{P})$ *fails to compute the longest plane tree.*

Proof. We construct a point set $\mathcal{P}$ consisting of 9 points to show the claim. The first three points of $\mathcal{P}$ are the vertices of an equilateral triangle. We assume that two of these vertices are on a horizontal line and denote by x the center of this triangle. Let r_0 be the circumradius of the triangle.

Inside of this triangle, we want to place the vertices of two smaller equilateral triangles, where again the smaller is contained in the larger one. Set $\alpha = 17°$, $r_1 = \frac{2}{3}r_0$ and $r_2 = \frac{1}{3}r_0$. We then place the vertices of the first of the additional triangles on the circle $\partial D(x, r_1)$ in a way that the vertices have an angle of α to the nearest angular bisector of the outer triangle. We place the vertices of the second additional triangle on the circle $\partial D(x, r_2)$ again with an angle to the nearest bisector of the outer triangle. However, this time, the angle is $\alpha/2$. This construction is visualized in Figure 29.

Now consider the tree on this point set depicted by the solid edges in Figure 30 on the left. Note that the green, blue and yellow edges are rotational symmetric. The edges depicted in purple in the tree on the left side of Figure 30 are representatives of the possible edges currently not in the tree, that connect the outer to the inner triangle. However, as they are either the smallest edges in the cycle or intersect some edge not in a cycle, the algorithm cannot choose ab to be one of those edges.

The possible edges connecting the outer to the middle triangle are depicted in red. The red edge pq is not in a cycle with the edge it crosses, and thus it is not a possible swap keeping planarity. For the second red edge and the possible edges connecting the middle to the inner triangle, it can similarly be seen that they are shorter than any edge of the current tree. Finally, for the edges along the triangles or the edges connecting the interior and the middle triangle, it can easily be verified that there will be no strictly smaller edge in the unique cycle closed by them.

On the other hand, in the tree depicted in Figure 30 on the right each pair of same colored edges is longer than its counterpart to the left. Therefore, `AlgLocal`$(\mathcal{P})$ does not return a longest plane spanning tree. □

6 Conclusions

We leave several open questions:

(1) What is the correct approximation factor of the algorithm `AlgSimple`$(\mathcal{P})$ presented in Section 2? While each single lemma in Section 2 is tight for some case, it is hard to believe that the whole analysis, leading to the approximation factor $f \doteq 0.5467$, is tight. We conjecture that the algorithm has a better approximation guarantee.

(2) What is the approximation factor BoundDiam(3) achieved by the polynomial time algorithm that outputs the longest plane tree with diameter 3? By Theorem 1.5 it is at most $5/6$ (and by [1] it is at least $1/2$).

(3) For a fixed diameter $d \geq 4$, is there a polynomial-time algorithm that outputs the longest plane tree with diameter at most d? By Theorem 1.6 we know the answer is yes when $d = 3$. And Theorem 1.7 gives a positive answer for special classes of trees with diameter 4. Note that a hypothetical **polynomial-time approximation scheme (PTAS)** has to consider trees of unbounded diameter because of Theorem 1.4. It is compatible with our current knowledge that computing an optimal plane tree of diameter, say, $O(1/\varepsilon)$ would give a PTAS.

(4) Is the general problem of finding the longest plane tree in P? A similar question can be asked for several other plane objects, such as paths, cycles, matchings, perfect matchings, or triangulations. The computational complexity in all cases is open.

Acknowledgments

We thank the anonymous referees for their careful reading of our submissions and for many helpful suggestions that have improved the readability of the manuscript.

References

[1] Noga Alon, Sridhar Rajagopalan, and Subhash Suri. 1995. Long non-crossing configurations in the plane. *Fundam. Inform.* 22, 4 (1995), 385–394. DOI: https://doi.org/10.3233/FI-1995-2245

[2] Greg Aloupis, Jean Cardinal, Sébastien Collette, Erik D. Demaine, Martin L. Demaine, Muriel Dulieu, Ruy Fabila-Monroy, Vi Hart, Ferran Hurtado, Stefan Langerman, et al. 2010. Matching points with things. In *Theoretical Informatics (LATIN '10)*. Alejandro López-Ortiz (Ed.), Lecture Notes in Computer Science, Vol. 6034, 456–467. DOI: https://doi.org/10.1007/978-3-642-12200-2_40

[3] Sanjeev Arora. 1998. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM* 45, 5 (1998), 753–782. DOI: https://doi.org/10.1145/290179.290180

[4] Sanjeev Arora and Kevin L. Chang. 2004. Approximation schemes for degree-restricted MST and red-blue separation problems. *Algorithmica* 40, 3 (2004), 189–210. DOI: https://doi.org/10.1007/s00453-004-1103-4

[5] Alexander I. Barvinok, Sándor P. Fekete, David S. Johnson, Arie Tamir, Gerhard J. Woeginger, and Russell Woodroofe. 2003. The geometric maximum traveling salesman problem. *J. ACM* 50, 5 (2003), 641–664. DOI: https://doi.org/10.1145/876638.876640

[6] Ahmad Biniaz. 2020. Euclidean bottleneck bounded-degree spanning tree ratios. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA '20)*. SIAM, 826–836. DOI: https://doi.org/10.1137/1.9781611975994.50

[7] Ahmad Biniaz. 2020. Improved approximation ratios for two Euclidean maximum spanning tree problems. arXiv:2010.03870. Retrieved from https://arxiv.org/abs/2010.03870

[8] Ahmad Biniaz, Prosenjit Bose, Kimberly Crosbie, Jean-Lou De Carufel, David Eppstein, Anil Maheshwari, and Michiel Smid. 2019. Maximum plane trees in multipartite geometric graphs. *Algorithmica* 81, 4 (2019), 1512–1534. DOI: https://doi.org/10.1007/s00453-018-0482-x

[9] Johannes Blömer. 1991. Computing sums of radicals in polynomial time. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS '91)*. IEEE Computer Society, 670–677. DOI: https://doi.org/10.1109/SFCS.1991.185434

[10] Johannes Blömer. 1991. Computing sums of radicals in polynomial time. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 670–677. DOI: https://doi.org/10.1109/SFCS.1991.185434

[11] Sergio Cabello, Aruni Choudhary, Michael Hoffmann, Katharina Klost, Meghana M. Reddy, Wolfgang Mulzer, Felix Schröder, and Josef Tkadlec. 2020. A better approximation for longest noncrossing spanning trees. *In Proceedings of the 36th European Workshop on Computational Geometry (EuroCG '20)*.

[12] Timothy M. Chan. 2004. Euclidean bounded-degree spanning tree ratios. *Discret. Comput. Geom.* 32, 2 (2004), 177–194. DOI: https://doi.org/10.1007/s00454-004-1117-3

[13] Francis Y. L. Chin, Jianbo Qian, and Cao An Wang. 2004. Progress on maximum weight triangulation. In *Proceedings of the 10th Annual International Conference on Computing and Combinatorics (COCOON '04)*, Lecture Notes in Computer Science, Vol. 3106. Springer, 53–61. DOI: https://doi.org/10.1007/978-3-540-27798-9_8

[14] Richard Cole and Chee-Keng Yap. 1984. Geometric retrieval problems. *Inf. Control*. 63, 1/2 (1984), 39–57. DOI: https://doi.org/10.1016/S0019-9958(84)80040-6

[15] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms* (3rd ed.). MIT Press. Retrieved from http://mitpress.mit.edu/books/introduction-algorithms

[16] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. 2008. *Computational Geometry: Algorithms and Applications* (3rd ed.). Springer-Verlag. DOI: https://doi.org/10.1007/978-3-540-77974-2

[17] Adrian Dumitrescu and Csaba D. Tóth. 2010. Long non-crossing configurations in the plane. *Discrete Comput. Geom.* 44, 4 (2010), 727–752. DOI: https://doi.org/10.1007/s00454-010-9277-9

[18] Alon Efrat, Alon Itai, and Matthew J. Katz. 2001. Geometry helps in bottleneck matching and related problems. *Algorithmica* 31, 1 (2001), 1–28. DOI: https://doi.org/10.1007/s00453-001-0016-8

[19] David Eppstein. 2000. Spanning trees and spanners. In *Handbook of Computational Geometry*. Jörg-Rüdiger Sack and Jorge Urrutia (Eds.), Elsevier, 425–461. DOI: https://doi.org/10.1016/b978-044482537-7/50010-3

[20] Andrea Francke and Michael Hoffmann. 2009. The Euclidean degree-4 minimum spanning tree problem is NP-hard. In *Proceedings of the 25th ACM Symposium on Computational Geometry*. ACM, 179–188. DOI: https://doi.org/10.1145/1542362.1542399

[21] Peter D. Gilbert. 1979. *New Results in Planar Triangulations*. Technical Report R–850. University of Illinois Coordinated Science Lab.

[22] Sariel Har-Peled. 2011. *Geometric Approximation Algorithms*. Mathematical Surveys and Monographs, Vol. 173. American Mathematical Society. DOI: https://doi.org/10.1090/surv/173

[23] Gheza Tom Klincsek. 1980. Minimal triangulations of polygonal domains. *Ann. Discrete Math.* 9 (1980), 121–123. DOI: https://doi.org/10.1016/S0167-5060(08)70044-X

[24] Jirí Matoušek. 1993. Range searching with efficient hierarchical cutting. *Discret. Comput. Geom.* 10 (1993), 157–182. DOI: https://doi.org/10.1007/BF02573972

[25] Joseph S. B. Mitchell. 1999. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput*. 28, 4 (1999), 1298–1309. DOI: https://doi.org/10.1137/S0097539796309764

[26] Joseph S. B. Mitchell. 2004. Shortest paths and networks. In *Handbook of Discrete and Computational Geometry* (2nd ed.), Jacob E. Goodman and Joseph O'Rourke (Eds.), Chapman and Hall/CRC, 607–641. DOI: https://doi.org/10.1201/9781420035315.ch27

[27] Joseph S. B. Mitchell and Wolfgang Mulzer. 2017. Proximity algorithms. In *Handbook of Discrete and Computational Geometry* (3rd. ed.), Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth (Eds.), CRC Press, Chapter 32, 849–874. DOI: https://doi.org/10.1201/9781315119601

[28] Wolfgang Mulzer. 2004. *Minimum Dilation Triangulations for the Regular n-Gon*. Master's thesis. Freie Universität Berlin, Germany.

[29] Wolfgang Mulzer and Johannes Obenaus. 2020. The tree stabbing number is not monotone. In *Proceedings of the 36th European Workshop on Computational Geometry (EWCG '20)*, 78:1–78:8.

[30] Wolfgang Mulzer and Günter Rote. 2008. Minimum-weight triangulation is NP-hard. *J. ACM* 55, 2 (2008), 11:1–11:29. DOI: https://doi.org/10.1145/1346330.1346336

[31] Giri Narasimhan and Michiel Smid. 2007. *Geometric Spanner Networks*. Cambridge University Press, Cambridge. DOI: https://doi.org/10.1017/CBO9780511546884

[32] Christos H. Papadimitriou. 1977. The Euclidean traveling salesman problem is NP-complete. *Theor. Comput. Sci.* 4, 3 (1977), 237–244. DOI: https://doi.org/10.1016/0304-3975(77)90012-3

[33] Christos H. Papadimitriou and Umesh V. Vazirani. 1984. On two geometric problems related to the traveling salesman problem. *J. Algorithms* 5, 2 (1984), 231–246. DOI: https://doi.org/10.1016/0196-6774(84)90029-4

[34] Franco P. Preparata and Michael Ian Shamos. 1985. *Computational Geometry. An Introduction*. Springer-Verlag, xii+390 pages. DOI: https://doi.org/10.1007/978-1-4612-1098-6

[35] Jan Remy and Angelika Steger. 2009. A quasi-polynomial time approximation scheme for minimum weight triangulation. *J. ACM* 56, 3 Article 15 (2009), 1–47. DOI: https://doi.org/10.1145/1516512.1516517

[36] Emo Welzl. 1992. On spanning trees with low crossing numbers. In *Data Structures and Efficient Algorithms*. Lecture Notes in Computer Science, Vol. 594. Springer, 233–249. DOI: https://doi.org/10.1007/3-540-55488-2_30

[37] David P. Williamson and David B. Shmoys. 2011. *The Design of Approximation Algorithms*. Cambridge University Press, Cambridge. DOI: https://doi.org/10.1017/CBO9780511921735

Received 30 April 2024; revised 21 May 2025; accepted 31 July 2025