Long plane trees

Sergio Cabello

Institute of Mathematics, Physics and Mechanics, Slovenia
Faculty of Mathematics and Physics, University of Ljubljana, Slovenia

Michael Hoffmann

Department of Computer Science, ETH Zürich, Switzerland

Katharina Klost

Institut für Informatik, Freie Universität Berlin, Germany

Wolfgang Mulzer

Institut für Informatik, Freie Universität Berlin, Germany

Josef Tkadlec

Department of Mathematics, Harvard University, USA

Abstract

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 T_{OPT} for $\mathcal{P}$ with maximum total Euclidean edge length $|T_{\text{OPT}}|$. Despite more than two decades of research, it remains open if this problem is NP-hard. Thus, previous efforts have focused on polynomial-time algorithms that produce plane trees whose total edge length approximates $|T_{\text{OPT}}|$. The approximate trees in these algorithms all have small unweighted diameter, typically three or 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 T_{ALG} with diameter at most four and $|T_{\text{ALG}}| \geq 0.546 \cdot |T_{\text{OPT}}|$. 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).

2012 ACM Subject Classification Theory of computation → Routing and network design problems; Theory of computation → Approximation algorithms analysis; Theory of computation → Computational geometry; Mathematics of computing → Trees

Keywords and phrases geometric network design, spanning trees, plane straight-line graphs, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.23

Related Version *Full Version*: `https://arxiv.org/abs/2101.00445` [11]

Funding *Sergio Cabello*: Supported by the Slovenian Research Agency (P1-0297, J1-9109, J1-8130, J1-8155, J1-1693, J1-2452).
Michael Hoffmann: Supported by the Swiss National Science Foundation within the collaborative DACH project *Arrangements and Drawings* as SNSF Project 200021E-171681.
Wolfgang Mulzer: Supported in part by ERC StG 757609.

1 Introduction

Geometric network design is a common and well-studied task in computational geometry and combinatorial optimization [18, 21, 24, 25]. In this family of problems, we are given a set $\mathcal{P}$ of points, 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, 15, 23, 28], the maximum length (bottleneck) edge [6, 17], the maximum degree [4, 12, 19, 31], the dilation [18, 26, 29], or the stabbing number [27, 33] 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 our considerations to paths, trees, or triangulations, respectively, we are faced with the venerable problems of finding an optimum TSP tour [21], a Euclidean minimum spanning tree [15], or a minimum weight triangulation [28] for $\mathcal{P}$. These three examples also illustrate the wide variety of complexity aspects that we may encounter in geometric network design problems: the Euclidean TSP is known to be NP-hard [30], but it admits a PTAS [3, 23]. On the other hand, it is possible to find a Euclidean minimum spanning tree for $\mathcal{P}$ in polynomial time [15] (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 [28], but the existence of a PTAS is still open; however, a QPTAS is known [32].

In this work, we are interested in the interaction of two specific requirements for a geometric network 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, Rajagopalan, and Suri [1], who considered 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 all three problems are 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 [14, Chapters 16.4, 23].

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 (i.e., no three points in $\mathcal{P}$ are collinear). Such a tree is necessarily spanning. The general position assumption was also used in previous work [1, 16]; without it, one should specify whether overlapping edges are allowed, an additional complication that we would like to avoid.

If $\mathcal{P}$ is in convex position, 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 convex point sets [20, 22]. 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 research has 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 [16] 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 set of candidates provides 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 the output of the algorithm with the length of a longest (not necessarily plane) spanning tree. Such a tree may be longer by a factor of up to $\pi/2 > 1.5$ 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 number in itself, the objective is to construct longest plane trees and thus it is better to compare the length of the constructed plane trees against the true optimum, that is, against the length of the 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 [10]. 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.** *For any finite point set $\mathcal{P}$ in general position (no three points collinear), we can compute in polynomial time a plane tree of Euclidean length at least $f \cdot |T_{\text{OPT}}|$, where $|T_{\text{OPT}}|$ 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 "guesses" a longest edge of T_{OPT} and then constructs six trees: four stars and two more 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 the constants in the proof.

Second, we characterize longest plane trees for convex point sets. 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 convex point set $\mathcal{P}$ is a *zigzagging caterpillar* if its edges split the convex hull of $\mathcal{P}$ into faces that are all triangles.

▶ **Theorem 2.** *If $\mathcal{P}$ is convex then every longest plane tree on $\mathcal{P}$ is a zigzagging caterpillar.*

▶ **Theorem 3.** *For any caterpillar C, there is a convex point set $\mathcal{P}$ such that the unique longest tree for $\mathcal{P}$ is isomorphic to C.*

In particular, the diameter of a (unique) longest plane tree is not bounded by any constant. As a consequence, we obtain an upper bound on the approximation factor BoundDiam(d) that can be achieved by a plane tree of diameter at most d.

▶ **Theorem 4.** *For any $d \geq 2$, there is a convex point set $\mathcal{P}$ so that every plane tree of diameter at most d on $\mathcal{P}$ is at most*

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

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

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

▶ **Theorem 5.** *For any $\varepsilon > 0$, there is a convex point set $\mathcal{P}$ such that every longest plane tree on $\mathcal{P}$ of diameter 3 is at most $(5/6) + \varepsilon$ times as long as a longest (general) plane tree.*

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 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 $\mathcal{O}(n^4)$ time.*

▶ **Theorem 7.** *For any set $\mathcal{P}$ of points in general position and any three specified points on the boundary of the convex hull of $\mathcal{P}$, we can compute in polynomial time a 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 $|T^3_{\text{OPT}}|$ of a longest plane tree of diameter at most three can be computed in polynomial time, we do not know the corresponding approximation factor BoundDiam(3). The best bounds we are aware of are $1/2 \leq$ BoundDiam(3) $\leq 5/6$. The lower bound follows from [1], the upper bound is from Theorem 5. We conjecture that $|T^3_{\text{OPT}}|$ actually gives a better approximation factor than the tree constructed in Theorem 1—but we are unable to prove this.

Fourth, a natural way to design an algorithm for the longest plane spanning tree problem is the following local search heuristic [34]: 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 Lemma 17 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 optimal 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 T_{OPT} be a plane (i.e., noncrossing) spanning tree on $\mathcal{P}$ with maximum Euclidean edge length. As the previous algorithms [1,7,8,10,16], we make extensive use of stars. The *star* S_p rooted at some point $p \in \mathcal{P}$ is the tree that connects p to all other points of $\mathcal{P}$.

We also need the notion of "flat" point sets. A point set $\mathcal{P}$ is *flat* if 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 $\varepsilon > 0$. For flat point sets, we can approximate the length of an edge by subtracting the x-coordinates of its endpoints: the error becomes arbitrarily small as $\varepsilon \to 0$. 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 An Improved Approximation Algorithm

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

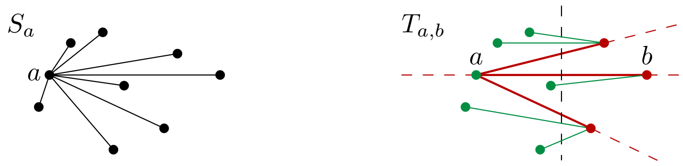


Figure 1 A tree S_a and a tree $T_{a,b}$.

the following trees $T_{a,b}$, for $a, b \in \mathcal{P}$ (see Figure 1): let $\mathcal{P}_a$ be the points of $\mathcal{P}$ closer to a than to b, and let $\mathcal{P}_b = \mathcal{P} \setminus \mathcal{P}_a$. First, connect a to every point in $\mathcal{P}_b$. Then, connect each point of $\mathcal{P}_a \setminus \{a\}$ to some point of $\mathcal{P}_b$ without introducing crossings. This yields a tree of diameter at most four. In general, $T_{a,b}$ and $T_{b,a}$ are different and neither is uniquely determined, but for $\mathcal{P}_a = \{a\}$ both $T_{a,b}$ and $T_{b,a}$ coincide with the star S_a.

Our algorithm `AlgSimple(`$\mathcal{P}$`)` computes all stars S_a and the tree $T_{a,b}$, for each ordered pair $a, b \in \mathcal{P}$, and it returns a longest one. The algorithm runs in polynomial time, as there are $\mathcal{O}(n^2)$ relevant trees, each of which can be found in polynomial time.

Given multiple trees that all contain a common edge ab, we direct all other edges towards ab and assign to each point in $\mathcal{P} \setminus \{a, b\}$ its unique outgoing edge. The edge ab remains undirected. Denote the length of the edge assigned to $p \in \mathcal{P} \setminus \{a, b\}$ in such a tree T by $\ell_T(p)$.

Theorem 1 states that for any $\mathcal{P}$, we have $|T_{\text{ALG}}| > 0.5467 \cdot |T_{\text{OPT}}|$. As a warm-up for the full proof, we first show a stronger result for the special case of flat point sets: if $\mathcal{P}$ is flat, we have $|T_{\text{ALG}}| \geq (2/3) \cdot |T_{\text{cr}}|$, where T_{cr} is a longest (possibly crossing) tree. In fact, the constant $2/3$ is tight when comparing to T_{cr}:

▶ **Observation 8.** *There is an infinite family of point sets $\mathcal{P}_1, \mathcal{P}_2, \dots$ with $|\mathcal{P}_n| = 2n$ and*

$$\lim_{n\to\infty} \frac{|T_{\text{OPT}}|}{|T_{\text{cr}}|} \leq \frac{2}{3}.$$

Proof. Let $\mathcal{P}_n = \{p_1, \dots, p_{2n}\}$ be a flat point set where the points p_i are spaced evenly on a convex arc with x-coordinates $1, \dots, 2n$, see Figure 2. It can be shown inductively, that the star S_{p_1} is a longest plane spanning tree and thus $|T_{\text{OPT}}| = |S_{p_1}| = \sum_{i=1}^{2n-1} i = 2n^2 - n$. On the other hand, the right side in Figure 2 shows a crossing spanning tree of total length $(2n-1) + 2\sum_{i=n}^{2n-2} i = 3n^2 - 3n + 1 \leq |T_{\text{cr}}|$. ◀

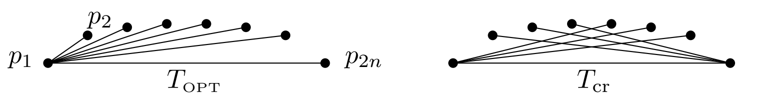


Figure 2 The point set $\mathcal{P}_n$ of $2n$ points with equally spaced x-coordinates $1, 2, \dots, 2n$, with a longest plane and the longest general spanning tree.

▶ **Theorem 9.** *Suppose $\mathcal{P}$ is flat. Then,*

$$|T_{\text{ALG}}| \geq \frac{2}{3}|T_{\text{cr}}| \geq \frac{2}{3}|T_{\text{OPT}}|.$$

Proof. As $|T_{\text{cr}}| \geq |T_{\text{OPT}}|$, it suffices to show the first inequality. Denote the diameter of $\mathcal{P}$ by ab (see Figure 3). Consider the four trees S_a, $T_{a,b}$, $T_{b,a}$, S_b. It suffices to show that there exists a $\beta \in (0, 1/2)$ such that

$$(1/2 - \beta)|S_a| + \beta|T_{a,b}| + \beta|T_{b,a}| + (1/2 - \beta)|S_b| \geq \frac{2}{3} \cdot |T_{\text{cr}}|.$$

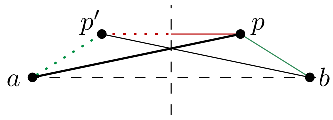


Figure 3 By triangle inequality and symmetry, we have $\|pp'\| + \|pb\| \geq \|p'b\| = \|pa\|$.

Here we fix $\beta = \frac{1}{3}$ and equivalently show:

$$\frac{|S_a| + 2|T_{a,b}| + 2|T_{b,a}| + |S_b|}{6} \geq \frac{2}{3} \cdot |T_{\text{cr}}| \tag{1}$$

which is enough, as

$$\max\{|S_a|, |T_{a,b}|, |T_{b,a}|, |S_b|\} \geq \frac{1}{6}(|S_a| + 2|T_{a,b}| + 2|T_{b,a}| + |S_b|)$$

The trees S_a, $T_{a,b}$, $T_{b,a}$, S_b all contain the edge ab, and since that edge realizes the diameter, we can assume that T_{cr} also contains ab. We fix a $p \in \mathcal{P} \setminus \{a, b\}$, assume without loss of generality that $\|pa\| \geq \|pb\|$, and denote by p' the reflection of p at the perpendicular bisector of ab (see Figure 3). Using the notation $\ell_T(p)$ from above,

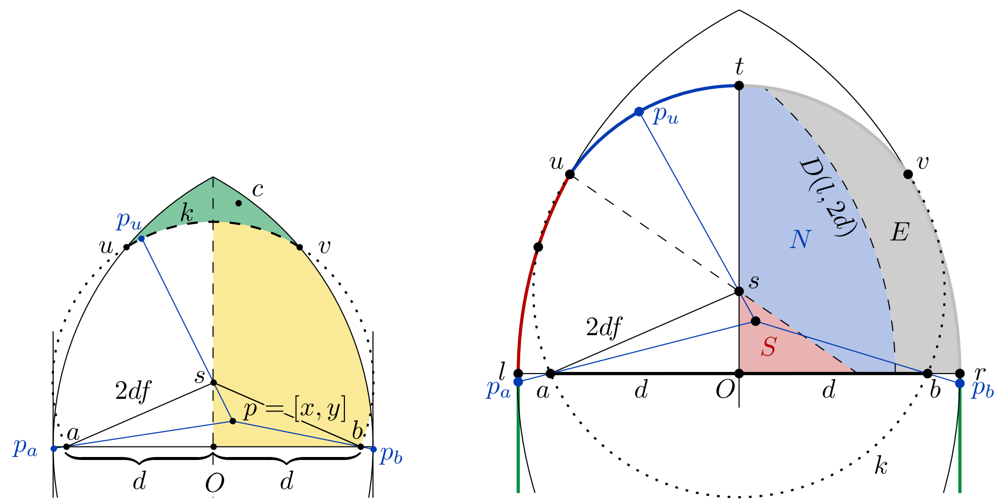
$$\begin{aligned}\frac{1}{6}\big(\ell_{S_a}(p) + 2\ell_{T_{a,b}}(p) + 2\ell_{T_{b,a}}(p) + \ell_{S_b}(p)\big) &\geq \frac{1}{6}\big(\|pa\| + 2\|pa\| + \|pp'\| + \|pb\|\big) \\ &\geq \frac{1}{6}\big(3\|pa\| + \|p'b\|\big) = \frac{2}{3} \cdot \|pa\| \geq \frac{2}{3} \cdot \ell_{T_{\text{cr}}}(p).\end{aligned}$$

Here, we used in the first step that $\ell_{S_a}(p) = \ell_{T_{a,b}}(p) = \|pa\|$, $\ell_{T_{b,a}}(p) \geq \|p'p\|/2$, and $\ell_{S_b}(p) = \|pb\|$. In the second and third step, we used the triangle inequality $\|pp'\| + \|pb\| \geq \|p'b\|$ and the symmetry $\|p'b\| = \|pa\|$. The final step follows since $\mathcal{P}$ is flat and hence $\ell_{T_{\text{cr}}}(p) \leq \max\{\|pa\|, \|pb\|\} = \|pa\|$. Now, (1) follows by summing over all $p \in \mathcal{P} \setminus \{a, b\}$. ◀

▶ **Theorem 1.** *For any finite point set $\mathcal{P}$ in general position (no three points collinear), we can compute in polynomial time a plane tree of Euclidean length at least $f \cdot |T_{\text{OPT}}|$, where $|T_{\text{OPT}}|$ 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 formally be stated later in this section. Without loss of generality, suppose $\mathcal{P}$ has diameter 2. Consider a longest edge ab of T_{OPT} and denote its length by $2d$ (we have $d \leq 1$).

Let $u, v \in \mathcal{P}$ be two points realizing the diameter of $\mathcal{P}$. Note that in general the longest edge of T_{OPT} does not realize the diameter and thus a, b and u, v differ. If $2df \leq 1$, it follows from previous work that one of S_u or S_v is long enough (see [10, Lemma 2.1]). Thus, we henceforth assume that $2df > 1$. Note that $\mathcal{P}$ lies in the lens $L = D(a, 2) \cap D(b, 2)$ and that the points a and b are in L. Choose a coordinate system with $a = (-d, 0)$ and $b = (d, 0)$, and let s, s' be the two points on the y-axis with $\|sa\| = \|sb\| = \|s'a\| = \|s'b\| = 2df$, where s is the point above the x-axis. Since $2df > 1$, the circles $k = \partial D(s, 2df)$ and $k' = \partial D(s', 2df)$ intersect the boundary of L. Let u, v and u', v' be the intersection points above and below the x-axis respectively, so that u and u' are to the left of the y-axis. The *far region* consists of the points in L above the arc of k between u and v in clockwise direction and of the points in L below the arc of k' between u' and v' in counter-clockwise direction. The *truncated lens* contains the remaining points, see Figure 4a.



(a) The lens is split into the far region (green) and the truncated lens

(b) The truncated lens is further subdivided into three regions E, N and S.

Figure 4 Subdivision of the lens.

In Lemma 10, we argue that if the far region contains a point $c \in \mathcal{P}$, then one of the three stars S_a, S_b, or S_c is long enough. Otherwise, if all of $\mathcal{P}$ lies in the truncated lens, we claim that one of the trees S_a, $T_{a,b}$, $T_{b,a}$, or S_b is long enough. These four trees all contain the edge ab. Thus, we can again use the notation $\ell_T(p)$ from above to define for any $p \in \mathcal{P} \setminus \{a, b\}$ and for any $\beta \in (0, 1/2)$, the weighted average

$$\text{avg}(p, \beta) = (1/2 - \beta) \cdot \ell_{S_a}(p) + \beta \cdot \ell_{T_{a,b}}(p) + \beta \cdot \ell_{T_{b,a}}(p) + (1/2 - \beta) \cdot \ell_{S_b}(p).$$

To finish the argument, we aim to find a $\beta \in (0, 1/2)$ so that for any $p \in \mathcal{P} \setminus \{a, b\}$, we have $\text{avg}(p, \beta) \geq f \cdot \ell_{T_{\text{OPT}}}(p)$ (note that $\ell_{T_{\text{OPT}}}(p)$ is defined, since ab is an edge of T_{OPT}). In contrast to the proof for Theorem 9, this now requires much more work. After that, the approximation guarantee follows by considering the sum $\sum_{p \in \mathcal{P} \setminus \{a,b\}} \text{avg}(p, \beta)$, as before.

For proving $\text{avg}(p, \beta) \geq f \cdot \ell_{T_{\text{OPT}}}(p)$, we can without loss of generality assume that $p = (x, y)$, with $x, y \geq 0$. The following definitions are illustrated in Figure 4a. Let p_a be the point with x-coordinate $-(2 - d)$ on the ray pa. If $x < d$, let p_b be the point with x-coordinate $2 - d$ on the ray pb. Otherwise, the ray pb does not intersect the vertical line with x-coordinate $2 - d$, and we set $p_b = b$. Additionally, define p_u to be the furthest point from p on the portion of the boundary of the far region that is contained in the circle $k = \partial D(s, 2df)$. The proof now proceeds in the following steps:

1. we show that $\ell_{T_{\text{OPT}}}(p) \leq \min\left\{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\right\}$ (Lemma 11);
2. we show that the term $\|pp_b\|$ in this upper bound can be omitted (Lemma 12);
3. we establish a lower bound on $\text{avg}(p, \beta)$ (Lemma 13); and
4. we use this lower bound to find constraints on β that ensure $\text{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_a\|\}$ and $\text{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_u\|\}$, respectively (Lemmas 14 and 15).

It then remains to show that there exists a β that satisfies both the constraints from Lemma 14 and from Lemma 15. It turns out that this holds for any $\beta \in (0, 1/2)$ with

$$(2f - 1) / \left(2\sqrt{5 - 8f} - 1\right) \leq \beta \leq 1 - f\sqrt{4f^2 - 1} - 2f^2. \quad (2)$$

Our choice of f ensures that the two expressions in (2) have the same value (≈ 0.1604). Setting β accordingly, we get the desired approximation (cf. the full version for the calculation). ◀

It remains to prove Lemmas 10 to 15. Their statements rely on the notation introduced in the proof outline of Theorem 1, so we recommend to first consult the paragraphs above.

▶ **Lemma 10.** *Let ab (with $\|ab\| = 2d$) be the longest edge of T_{OPT}. If $\mathcal{P}$ contains a point c in the far region, then* $\max\{|S_a|, |S_b|, |S_c|\} \geq f \cdot |T_{\text{OPT}}|$.

Proof. By the definition of the far region, the triangle abc is acute-angled and its circumradius R satisfies $R \geq 2df$. Let $g = \frac{1}{|\mathcal{P}_0|}\sum_{p\in\mathcal{P}_0} p$ be the center of mass of the point set $\mathcal{P}_0 \equiv \mathcal{P} \setminus \{a, b, c\}$, see Figure 5. Since the triangle abc is acute-angled, it has a vertex v with $\|vg\| \geq R$. By definition of g, we have $\sum_{p\in\mathcal{P}_0} \overrightarrow{vp} = |\mathcal{P}_0| \cdot \overrightarrow{vg}$, and the triangle inequality

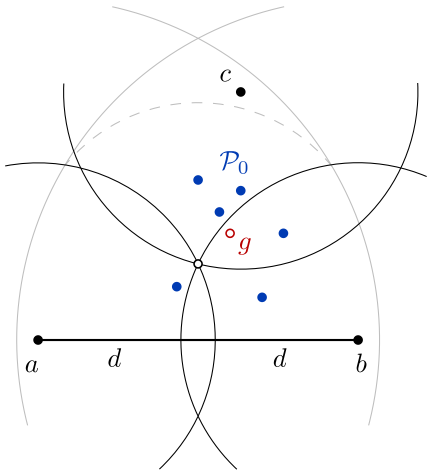


Figure 5 Lemma 10. In the illustration, $\mathcal{P}_0$ consists of 6 points and we can take $v = a$. The common point of the three black circles is the circumcenter of triangle abc.

gives $\sum_{p\in\mathcal{P}_0} \|vp\| \geq |\mathcal{P}_0| \cdot \|vg\| \geq (n-3) \cdot R$. As $\|va\| + \|vb\| + \|vc\| \geq 2R$ holds in any acute-angled triangle, we obtain $|S_v| \geq (n-1) \cdot R \geq (n-1) \cdot 2df \geq f \cdot |T_{\text{OPT}}|$. ◀

▶ **Lemma 11.** *For every point $p = (x, y)$ with $x, y \geq 0$ in the truncated lens, we have* $\ell_{T_{\text{OPT}}}(p) \leq \min\{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}\}$.

Proof Sketch. (Full proof in the full version) Let $l = (d-2, 0)$ and $r = (2-d, 0)$ be the left– and rightmost points of $D(a, 2) \cap D(b, 2)$. We divide the truncated lens into further regions (see Figure 4b): the region E lies inside the truncated lens but outside of $D(l, 2d)$, and the remainder of the truncated lens is divided into the part N above the line us and the part S below us. If $p \in E$, then $\min\{2d, \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\} = 2d$, and we are done, since $\ell_{T_{\text{OPT}}(p)} \leq 2d$. Next, assume that $p \in N \cup S$, and let p_f be the furthest point from p in the truncated lens. An exhaustive case distinction over the quadrant containing p_f shows that $\|pp_f\| \leq \max\{\|pp_a\|, \|pp_b\|, \|pp_u\|\}$, which proves the lemma. ◀

This bound can be simplified by using the following lemma:

▶ **Lemma 12.** *For every point $p = (x, y)$ with $x, y \geq 0$ in the truncated lens, if $\|pp_a\| \leq 2d$, then $\|pp_b\| \leq \|pp_a\|$.*

The algebraic proof can be found in the full version.

Now we give a general lower bound on $\text{avg}(p, \beta)$ that we will use in Lemmas 14 and 15.

▶ **Lemma 13.** *Let $p = (x, y) \in \mathbb{R}^2$ be a point with $x, y \geq 0$, and let $\beta \in (0, 1/2)$. Then,*

$$\mathrm{avg}(p, \beta) \geq \frac{d \cdot (1-\beta) + x \cdot 2\beta}{d + x} \cdot \|pa\|.$$

Proof Sketch. (Full proof in the full version) We expand the definition and replace the $\ell_T(p)$-terms by $\|pa\|$, $\|pb\|$, and x, respectively. By similar geometric arguments as in Theorem 9,

$$\mathrm{avg}(p, \beta) \geq (1/2) \cdot \|pa\| + (\beta/2) \cdot \|pa\| + ((1/2) - (3/2)\beta) \cdot \|pb\|.$$

Using $\|pb\| \geq \frac{d-x}{d+x} \cdot \|pa\|$, we get the desired

$$\mathrm{avg}(p, \beta) \geq \frac{(1+\beta)(d+x) + (1-3\beta)(d-x)}{2(d+x)} \cdot \|pa\| = \frac{(1-\beta) \cdot d + 2\beta \cdot x}{d+x} \cdot \|pa\|. \quad ◀$$

▶ **Lemma 14.** *Let $p = (x, y)$ be any point in the truncated lens with $x, y \geq 0$. Then, if $\frac{2f-1}{5-8f} \leq \beta \leq \frac{1}{2} \cdot f$, we have $\mathrm{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_a\|\}$.*

Proof Sketch. (Full proof in the full version) We show that if $x \geq 3d - 2$, then $\mathrm{avg}(p, \beta) \geq f \cdot 2d$, and if $x \leq 3d - 2$, then $\mathrm{avg}(p, \beta) \geq f \cdot \|pp_a\|$. Using Lemma 13, both cases reduce to the following inequality, which holds by the assumption on β:

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

▶ **Lemma 15.** *Let $p = (x, y)$ be any point in the truncated lens with $x, y \geq 0$. Suppose that $\beta < \frac{151}{304} \cdot f$ and that $\frac{1}{2} \leq f \leq \frac{19}{32}$, then $\mathrm{avg}(p, \beta) \geq f \cdot \min\{2d, \|pp_u\|\}$, if*

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

Proof Sketch. (Full proof in the full version) By Lemma 13, it suffices to show that

$$\lambda = \frac{d \cdot (1-\beta) + x \cdot 2\beta}{d+x} \cdot \|pa\| \geq f \cdot \min\{2d, \|pp_u\|\}. \tag{3}$$

Case 1: $y \leq y(u)$. λ is an increasing function in y and $\|pp_u\|$ is a decreasing function in y, for $y \leq y(u)$. Thus, it suffices to show (3) for $y = 0$. In this case, λ becomes $\lambda_0 = d \cdot (1-\beta) + x \cdot 2\beta$, which is positive. Let $q = (q_x, 0), q_x \geq 0$, be the point on the x-axis with $\|qs\| = 2d(1-f)$. For $p = q$, we have $\|pp_u\| \leq \|ps\| + \|sp_u\| = 2d$.

Case 1a: $0 \leq x \leq q_x$. The Pythagorean theorem and the bounds on β yield $\lambda_0 \geq f \cdot \|pp_u\|$.

Case 1b: $q_x < x$. It suffices to show $\lambda_0 \geq f \cdot 2d$, for $x = q_x$. This follows from Case 1a.

Case 2: $y > y(u)$ Now, $\|pp_u\| = \|pu\| \leq \|uv\|$. Also, we have $x(u) \geq -d$ and $x \leq d$, which gives $\min\{2d, \|pp_u\|\} = \|pp_u\|$. Thus, (3) becomes $\lambda \geq f \cdot \|pp_u\|$. From $y > y(u)$, we get $\|pa\| \geq \|pp_u\|$, so we need $\lambda/\|pa\| \geq f$. This follows by straightforward algebra. ◀

3 Convex and flat convex point sets

We present two results for convex point sets: (i) if $\mathcal{P}$ is convex, any longest plane tree is a caterpillar, and any caterpillar appears as the unique longest plane tree of a convex point set; and (ii) by looking at suitable flat convex sets, we prove upper bounds on the approximation factor achieved by the longest plane tree among those with diameter at most d.

Convex sets 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. We consider trees that span a given convex point set $\mathcal{P}$. We call (a drawing of) such a tree T a *zigzagging caterpillar* if T is a caterpillar and the *dual graph* $T^\star$ of T is a path, where $T^\star$ is defined as follows: consider a smooth closed curve through all points of $\mathcal{P}$. The curve bounds a convex region that is split by the $n-1$ edges of T into n subregions. Then $T^\star$ has a node for each such subregion and two nodes are connected if their subregions share an edge of T (see Figure 6).

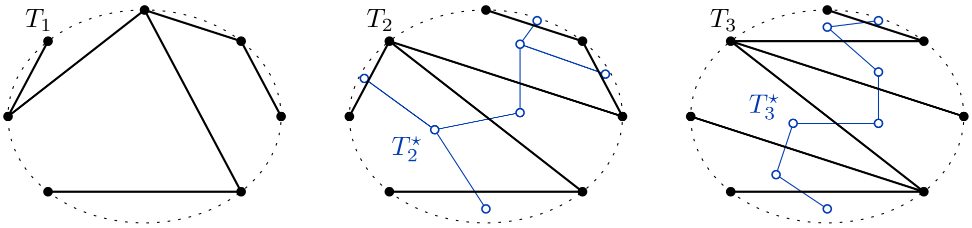


Figure 6 T_1 is spanning $\mathcal{P}$ but it is not a caterpillar. T_2 is a caterpillar but it is not zigzagging. T_3 is a zigzagging caterpillar, since the dual tree $T_3^\star$ is a path.

▶ **Theorem 2.** *If $\mathcal{P}$ is convex then every longest plane tree on $\mathcal{P}$ is a zigzagging caterpillar.*

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

Note that as $\mathcal{P}$ is assumed to be convex in this context, an optimal caterpillar can be found by applying the dynamic programming approach for the convex case described in Section 4.

Conversely, for every caterpillar C we construct a convex set $\mathcal{P}_C$ whose longest plane tree is isomorphic to C. In fact, $\mathcal{P}_C$ will be a *flat arc*: a flat convex point set $\{a_i = (x_i, y_i)\}_{i=1}^{m+1}$, where $x_i < x_j$, for $i < j$. The sequence $G(\mathcal{P}_C) = \{g_i\}_{i=1}^m = \{|x_{i+1} - x_i|\}_{i=1}^m$ is the *gap sequence* of $\mathcal{P}_C$. Given a spanning tree T for $\mathcal{P}_C$, we define its *cover sequence* $\text{Cov}(T) = \{c_i\}_{i=1}^m$ where c_i denotes the number of times gap g_i is "covered", see Figure 7. Then, $|T| = \sum_{i=1}^m c_i \cdot g_i$.

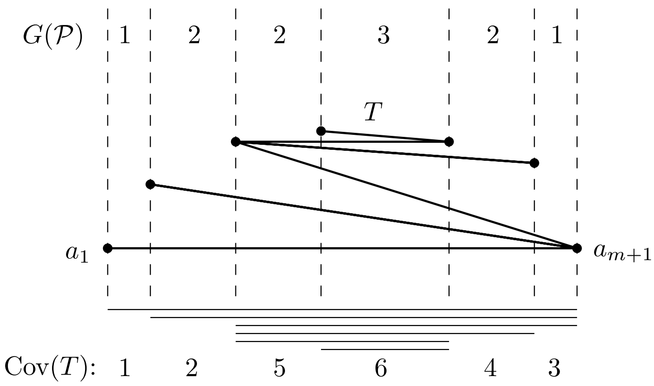


Figure 7 A tree with its gap and cover sequence.

▶ **Lemma 16.** *Consider a flat arc $\{a_1, \dots, a_{m+1}\}$ and a zigzagging caterpillar T containing the edge $a_1 a_{m+1}$. Then the sequence $\mathrm{Cov}(T)$ is a unimodal permutation of $\{1, 2, \dots, m\}$.*

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, $a_1 a_{m+1}$ is an edge of T, either $a_1 a_m$ or $a_2 a_{m+1}$ is an edge of T too. Without loss of generality assume $a_1 a_m$ is an edge of T. Then $T \setminus \{a_1 a_{m+1}\}$ is a zigzagging caterpillar on m points $a_1, \dots, a_m$ containing the edge $a_1 a_m$, hence by induction its cover sequence is a unimodal permutation of $\{1, 2, \dots, m-1\}$. Adding the omitted edge $a_1 a_{m+1}$ adds 1 to each of the $m - 1$ elements and appends a 1 to the list, giving rise to a unimodal permutation of $\{1, 2, \dots, m\}$. This completes the proof. ◀

▶ **Theorem 3.** *For any caterpillar C, there is a convex point set $\mathcal{P}$ such that the unique longest tree for $\mathcal{P}$ is isomorphic to C.*

Proof. Consider a flat arc $\mathcal{P} = \{a_1, \dots, a_{m+1}\}$, with a yet unspecified gap sequence $\{g_i\}_{i=1}^m$, and let T be a drawing of C onto $\mathcal{P}$ that contains the edge $a_1 a_{m+1}$ and is zigzagging (such a drawing always exists). By Lemma 16, the cover sequence $\mathrm{Cov}(T) = \{c_i\}_{i=1}^m$ is a unimodal permutation of $\{1, 2, \dots, m\}$. The total length of T can be expressed as $|T| = \sum_{i=1}^m c_i \cdot g_i$.

Now we specify the gap sequence: for $i = 1, \dots, m$, set $g_i = c_i$. It remains to show that T constitutes the longest plane tree T_{OPT} of $\mathcal{P}$.

By Theorem 2, T_{OPT} is a zigzagging caterpillar. Also, $a_1 a_{m+1}$ is an edge of T_{OPT}: suppose not. Since $a_1 a_{m+1}$ 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 $a_1 a_{m+1}$, so omitting any other edge from C yields a longer plane tree, a contradiction. We can thus apply Lemma 16 to see that $\mathrm{Cov}(T_{\text{OPT}})$ is a unimodal permutation π of $\{1, 2, \dots, m\}$ 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 permuations, the Cauchy-Schwarz inequality gives

$$|T_{\text{OPT}}| = \sum_{i=1}^{m} \pi_i \cdot g_i \leq \sqrt{\sum_{i=1}^{m} \pi_i^2 \cdot \sum_{i=1}^{m} g_i^2} = \sum_{i=1}^{m} c_i^2 = |T|, \text{with equality iff } \pi_i = c_i, \text{ for all } i. \quad (4)$$

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

Upper bounds on $\mathrm{BoundDiam}(d)$**.** The algorithms for approximating $|T_{\text{OPT}}|$ often produce trees with small diameter. Given $d \geq 2$ and a point set $\mathcal{P}$, let $T^d_{\text{OPT}}(\mathcal{P})$ be a longest plane tree on $\mathcal{P}$ among those with diameter at most d. We ask what is the approximation ratio

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

For $d = 2$, this question concerns the performance of stars. A result of Alon, Rajagopalan, and Suri [1, Theorem 4.1] can be restated as $\mathrm{BoundDiam}(2) = 1/2$. We show a crude upper bound on $\mathrm{BoundDiam}(d)$ for general d and a specific upper bound for the case $d = 3$. (Note that Theorem 6 shows that for any fixed $\mathcal{P}$ we can compute $|T^3_{\text{OPT}}(\mathcal{P})|$ in polynomial time.) Our proofs use the notions of flat arc, gap sequence, and cover sequence defined above.

▶ **Theorem 4.** *For any $d \geq 2$, there is a convex point set $\mathcal{P}$ so that every plane tree of diameter at most d on $\mathcal{P}$ is at most*

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

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

Proof. Let $\mathcal{P}$ be a flat arc on $d+2$ points with gap sequence $G = (1, 3, 5, \ldots, d+1, \ldots, 6, 4, 2)$. Since G is unimodal, we can argue as in the proof of Theorem 3 to see that T_{OPT} is the zigzagging caterpillar whose cover sequence is G, i.e., a path with $d+1$ edges (and diameter $d+1$). Moreover, this path is the only optimal plane tree spanning the flat arc because of Theorem 2 and the Cauchy-Schwarz inequality; see the argument leading to (4). Therefore, any other plane spanning tree $T \neq T_{\text{OPT}}$, zigzagging caterpillar or not, has an integer length less than $|T_{\text{OPT}}|$. Using $|T_{\text{OPT}}| = \sum_{i=1}^{d+1} i^2 = \frac{1}{6}(d+1)(d+2)(2d+3) = \frac{1}{3}d^3 + o(d^3)$, we obtain

$$\text{BoundDiam}(d) \leq \frac{|T_{\text{OPT}}| - 1}{|T_{\text{OPT}}|} = 1 - \frac{6}{(d+1)(d+2)(2d+3)} = 1 - \Theta(1/d^3). \quad ◀$$

For $d = 3$, Theorem 4 gives $\text{BoundDiam}(3) \leq 29/30$. By tailoring the point set size, the gap sequence $\{g_i\}_{i=1}^m$, and by considering non-arcs, we improve this to $\text{BoundDiam}(3) \leq 5/6$.

▶ **Theorem 5.** *For any $\varepsilon > 0$, there is a convex point set $\mathcal{P}$ such that every longest plane tree on $\mathcal{P}$ of diameter 3 is at most $(5/6) + \varepsilon$ times as long as a longest (general) plane tree.*

Proof (Sketch). (Full proof in the full version) Let $\mathcal{P}_{4k+2}$ consist of two flat arcs, symmetric with respect to a horizontal line, 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 opposing points, four unit-spaced arcs of k points each, and a large central gap of length $2k+1$ (see Figure 8).

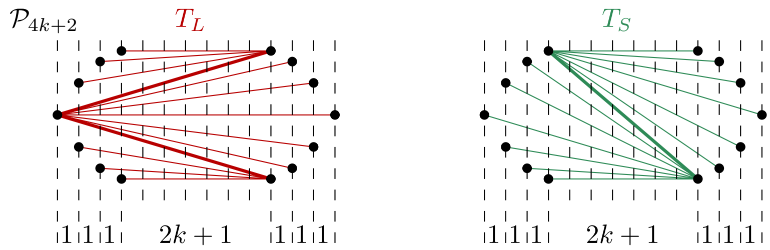


Figure 8 An illustration of the point set $\mathcal{P}_{4k+2}$ when $k = 3$, with trees T_L (red) and T_S (green).

On the one hand, straightforward counting gives $|T_{\text{OPT}}| \geq |T_L| = 12k^2 + 6k + 1$, where T_L is the tree depicted in Figure 8. On the other hand, any tree with diameter at most 3 is either a star or it contains an edge ab such that each other point of $\mathcal{P}$ is connected either to a or to b. For a star T, simple computation gives $|T| \leq 8k^2 + 6k + 1$. For the other case, one can show that the longest tree is obtained when points a, b lie on the opposite sides of the large central gap and at least one of them lies on the boundary of this gap, as is the case for instance for the tree T_S depicted in Figure 8. We have $|T_S| = 10k^2 + 6k + 1$, thus

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

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

4 Polynomial time algorithms for small diameter

We show how to compute a longest tree of diameter at most three in polynomial time, using dynamic programming. The main challenge is to devise an appropriate partition into independent subproblems. Our approach bears some resemblance to the polynomial time

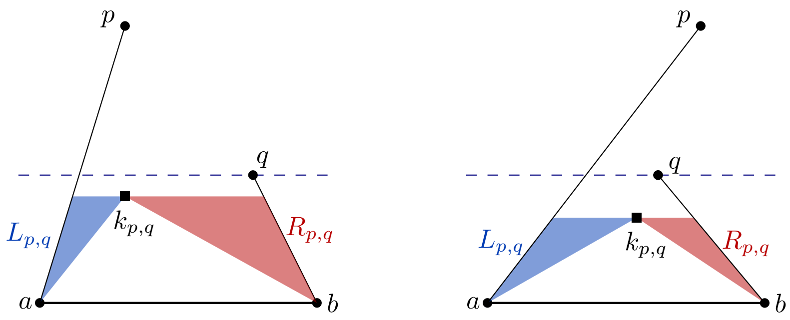


Figure 9 Fixing $k_{p,q}$ gives two possible triangular regions where edges are forced.

plane matching algorithm of Aloupis et al. [2]. The main challenge in our case is the efficient implementation of the dynamic program.

Our approach extends to a certain class of diameter-four trees, see the full version of this paper. Every tree of diameter two or three is a *bistar*, that is, it contains two vertices a and b so that every edge is incident to at least one of a or b. To prove Theorem 6, we note that there are $\Theta(n^2)$ choices for a and b, and we show how to compute a longest bistar rooted at a fixed pair a, b in $\mathcal{O}(n^2)$ time.

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. As no edge will cross this line, we can also assume that all points lie above this line.

The subproblems for the dynamic program are indexed by ordered pairs p, q of distinct points from $\mathcal{P}$, so that the line segments ap and bq do not cross. A pair that satisfies this condition is a *valid pair*. For each valid pair p, q, 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 $y = \min\{y(p), y(q)\}$. We define the value $Z(p, q)$ as 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 $Q(p, q)$. If we connect $k_{p,q}$ 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 force the triangle $R_{p,q}$ defined by bq, $bk_{p,q}$ and the line $y = y(k_{p,q})$; see Figure 9. 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 the valid pair $p, k_{p,q}$. This yields the following recurrence for each valid pair p, q:

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

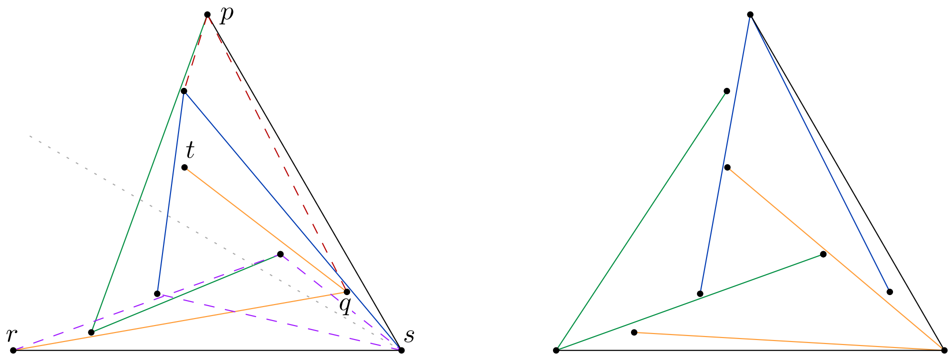
Using the values $Z(p, q)$, for all valid p, q, together with a specialized approach to solve the relevant range searching problems, we can show that a longest plane bistar for a fixed pair a, b of vertices can be computed in $\mathcal{O}(n^2)$ time; for details see the full version.

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 [34]. 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 spanning tree:
   a. Set $T \to T \cup \{ab\} \setminus \{cd\}$. `// tree` $T \cup \{ab\} \setminus \{cd\}$ `is longer than` T
3. Output T.



(a) A tree which cannot be locally improved.

(b) A tree where each pair of edges in the same color is at least as long as the matching pair in Figure 10a.

Figure 10 The algorithm `AlgLocal(`$\mathcal{P}$`)` can get stuck.

▶ **Lemma 17.** *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 points are placed on three concentric equilateral which are slightly rotated, see Figure 10.

Now consider the tree on this point set depicted by the solid edges in Figure 10a. Note that the green, blue and yellow edges are rotational symmetric. A simple case distinction, using the dashed edges as prototype for different non-edges shows that `AlgLocal(`$\mathcal{P}$`)` stops at this tree. On the other hand, in the tree depicted in Figure 10b each pair of the same colored edges is longer than its counterpart in Figure 10a. Therefore `AlgLocal(`$\mathcal{P}$`)` does not yield a correct result. ◀

We remark that point sets with the same property exist on any number $n \geq 9$ of points: it suffices to (repeatedly) duplicate the edge qt and perturb its endpoint t.

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 5 it is at most 5/6 (and by [1] it is at least 1/2).
3. For a fixed $d \geq 4$, is there a polynomial-time algorithm that outputs the longest plane tree with diameter at most d? By Theorem 6 we know the answer is yes when $d = 3$. And Theorem 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 4. It is compatible with our current knowledge that computing an optimal plane tree of diameter, say, $\mathcal{O}(1/\varepsilon)$ 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.

References

1 Noga Alon, Sridhar Rajagopalan, and Subhash Suri. Long non-crossing configurations in the plane. *Fundam. Inform.*, 22(4):385–394, 1995. `doi: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, Maria Saumell, Carlos Seara, and Perouz Taslakian. Matching points with things. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics*, volume 6034, pages 456–467, 2010. URL: `http://link.springer.com/10.1007/978-3-642-12200-2_40`, `doi:10.1007/978-3-642-12200-2_40`.

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

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

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

6 Ahmad Biniaz. Euclidean bottleneck bounded-degree spanning tree ratios. In *Proc. 31st Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 826–836, 2020. `doi:10.1137/1.9781611975994.50`.

7 Ahmad Biniaz. Improved approximation ratios for two Euclidean maximum spanning tree problems. `arXiv:2010.03870`, 2020.

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

9 Johannes Blömer. Computing sums of radicals in polynomial time. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 670–677, 1991. `doi:10.1109/SFCS.1991.185434`.

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

11 Sergio Cabello, Michael Hoffmann, Katharina Klost, Wolfgang Mulzer, and Josef Tkadlec. Long plane trees. *arXiv preprint arXiv:2101.00445*, 2021.

12 Timothy M. Chan. Euclidean bounded-degree spanning tree ratios. *Discrete Comput. Geom.*, 32(2):177–194, 2004. URL: `http://www.springerlink.com/index/10.1007/s00454-004-1117-3`.

13 Francis Y. L. Chin, Jianbo Qian, and Cao An Wang. Progress on maximum weight triangulation. In *Proc. 10th Annu. Int. Conf. Computing and Combinatorics (COCOON)*, pages 53–61, 2004. `doi:10.1007/978-3-540-27798-9\_8`.

14 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT Press, 3rd edition, 2009. URL: `http://mitpress.mit.edu/books/introduction-algorithms`.

15 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational geometry. Algorithms and applications.* Springer-Verlag, Berlin, third edition, 2008. `doi:10.1007/978-3-540-77974-2`.

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

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

18 David Eppstein. Spanning trees and spanners. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. North Holland / Elsevier, 2000. `doi:10.1016/b978-044482537-7/50010-3`.

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

20 P. D. Gilbert. New results in planar triangulations. Technical Report R–850, Univ. Illinois Coordinated Science Lab, 1979.

21 Sariel Har-Peled. *Geometric approximation algorithms*, volume 173 of *Mathematical Surveys and Monographs.* American Mathematical Society, Providence, RI, 2011. `doi:10.1090/surv/173`.

22 Gheza Tom Klincsek. Minimal triangulations of polygonal domains. *Ann. Discrete Math.*, 9:121–123, 1980.

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

24 Joseph S. B. Mitchell. Shortest paths and networks. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 607–641. Chapman and Hall/CRC, 2nd edition, 2004. `doi:10.1201/9781420035315.ch27`.

25 Joseph S. B. Mitchell and Wolfgang Mulzer. Proximity algorithms. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 32, pages 849–874. CRC Press, Boca Raton, 3rd edition, 2017. `doi:10.1201/9781315119601`.

26 Wolfgang Mulzer. Minimum dilation triangulations for the regular n-gon. Master's thesis, Freie Universität Berlin, Germany, 2004.

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

28 Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is NP-hard. *J. ACM*, 55(2):11:1–11:29, 2008. `doi:10.1145/1346330.1346336`.

29 Giri Narasimhan and Michiel Smid. *Geometric spanner networks.* Cambridge University Press, Cambridge, 2007. `doi:10.1017/CBO9780511546884`.

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

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

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

33 Emo Welzl. On spanning trees with low crossing numbers. In *Data structures and efficient algorithms*, volume 594 of *Lecture Notes in Comput. Sci.*, pages 233–249. Springer, Berlin, 1992. doi:10.1007/3-540-55488-2_30.

34 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. doi:10.1017/CBO9780511921735.