



HAL
open science

Euclidean Bottleneck Steiner Tree is Fixed-Parameter Tractable

Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh,
Jie Xue

► **To cite this version:**

Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, Jie Xue. Euclidean Bottleneck Steiner Tree is Fixed-Parameter Tractable. SODA 2024 - 35th ACM/SIAM Symposium on Discrete Algorithms, Jan 2024, Alexandria, VA, United States. pp.699-711, 10.1137/1.9781611977912.27 . hal-04778878

HAL Id: hal-04778878

<https://hal.science/hal-04778878v1>

Submitted on 12 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Euclidean Bottleneck Steiner Tree is Fixed-Parameter Tractable

Sayan Bandyopadhyay* William Lochet† Daniel Lokshтанov‡
Saket Saurabh§ Jie Xue¶

Abstract

In the Euclidean Bottleneck Steiner Tree problem, the input consists of a set of n points in \mathbb{R}^2 called terminals and a parameter k , and the goal is to compute a Steiner tree that spans all the terminals and contains at most k points of \mathbb{R}^2 as Steiner points such that the maximum edge-length of the Steiner tree is minimized, where the length of a tree edge is the Euclidean distance between its two endpoints. The problem is well-studied and is known to be NP-hard. In this paper, we give a $k^{O(k)}n^{O(1)}$ -time algorithm for Euclidean Bottleneck Steiner Tree, which implies that the problem is fixed-parameter tractable (FPT). This settles an open question explicitly asked by Bae et al. [Algorithmica, 2011], who showed that the ℓ_1 and ℓ_∞ variants of the problem are FPT. Our approach can be generalized to the problem with ℓ_p metric for any rational $1 \leq p \leq \infty$, or even other metrics on \mathbb{R}^2 .

1 Introduction

Given a (finite) set P of points in \mathbb{R}^2 , a *Steiner tree* on P refers to a tree with node set $P \cup S$ for some finite set $S \subseteq \mathbb{R}^2$ called *Steiner points*. The length of an edge in a Steiner tree is defined as the Euclidean distance between its two endpoints. In the *Euclidean Steiner tree* (EST) problem, we are given a set P of n points in \mathbb{R}^2 , and our goal is to compute a Steiner tree on P such that the total length of the tree edges is minimized. The EST problem is a classical combinatorial optimization problem whose history dates back to the 19th century (see [12] for more details). Motivated by both theoretical interest and practical applications, EST and its variants have been studied extensively in the past decades (see [19, 27] for applications to network problems, and [25] for a compendium of many Steiner tree problems). The seminal work of Arora [4] and Mitchell [34] independently gave polynomial-time approximation schemes (PTAS) for the EST problem. Besides the Euclidean setting, the Steiner tree problem can also be defined in general graphs, where we are given an edge-weighted graph G and a set of *terminal* vertices, and the goal is to find a minimum-weight tree in G which connects all terminal vertices. The Steiner tree problem in the graph setting was considered as one of the most fundamental NP-complete problems in the seminal paper of Karp [28], and has been extensively studied in the context of parameterized algorithms [10, 16, 17, 18, 21, 35]; we shall briefly summarize the work on this topic at the end of this section.

In this paper, we study a min-max version of the EST problem, called *Euclidean Bottleneck Steiner Tree* (EBST), which originates back to the late 80s and has several applications in VLSI design, facility location, and communication networks [15, 20].

*Portland State University, USA. Email: sayanb@pdx.edu.

†LIRMM, Université de Montpellier, CNRS, Montpellier, France. Email: william.lochet@gmail.com.

‡University of California, Santa Barbara, USA. Email: daniello@ucsb.edu.

§Institute of Mathematical Sciences, Chennai, India. Email: saket@imsc.res.in.

¶New York University Shanghai, China. Email: jiexue@nyu.edu.

EUCLIDEAN BOTTLENECK STEINER TREE**Parameter:** k **Input:** A set P of n points in \mathbb{R}^2 and an integer k .**Output:** Compute a Steiner tree on P with at most k Steiner points such that the maximum length of the tree edges is minimized.

Note that in the EBST problem, the restriction k on the number of Steiner points is necessary (while this is not the case in the EST problem), because without such a restriction, we can have a Steiner tree in which the length of every edge is arbitrarily small. (In fact, the EST problem with a restricted number k of Steiner points has also been studied. Brazil et al. [13] gave a more general $O(n^k)$ time algorithm for this variant of EST.)

The EBST problem has received much attention over years [5, 6, 14, 15, 20, 22, 31, 38, 39]. It is known to be NP-hard, and even hard to approximate up to a factor of $\sqrt{2}$ [20]. On the positive side, a polynomial-time 1.866-approximation algorithm for EBST was known [39]. Furthermore, different variants of EBST have been considered, including the full-tree version which requires all points in P to be the leaves of the Steiner tree [1, 9], the bichromatic version [2], the variant which prohibits edges between Steiner points [30], etc. A closely related problem, Euclidean Bottleneck Steiner Path, has also been studied [3, 26].

Even though approximation algorithms were known for EBST in full generality (as well as for some special cases) [22, 30, 38, 39], no exact algorithm was known for over more than one decade. Bae et al. [5, 6] initiated the study on the exact algorithms for EBST. They first observed that when $k \leq 2$, EBST is polynomial-time solvable [6]. Then in the follow-up work [5], they gave an $n^{O(k)}$ -time algorithm for EBST, showing that the problem is polynomial-time solvable for any fixed k . However, this does not settle the fixed-parameter tractability of the problem. Bae et al. explicitly asked the following open question in [5].

Question: Is EUCLIDEAN BOTTLENECK STEINER TREE fixed-parameter tractable (FPT), i.e., does it admit an algorithm with running time $f(k) \cdot n^{O(1)}$?

On the positive side, Bae et al. [5] already showed that the ℓ_1 variant (or equivalently the ℓ_∞ variant) of EBST is FPT. However, as is typical in computational geometry, the problem under ℓ_1 metric is substantially easier than that under Euclidean metric, as the ℓ_1 -disks are axis-parallel squares (rotated by an angle of $\frac{\pi}{4}$), which are usually much more tractable. In fact, the FPT algorithm given by Bae et al. [5] heavily relies on the geometry of squares (or more generally, rectilinear domains), and thus they failed to extend the algorithm to EBST. For more than one decade, the above question has remained open.

In this paper, we resolve the above question by designing the first FPT algorithm for EBST. Specifically, we obtain the following result.

Theorem 1. *EUCLIDEAN BOTTLENECK STEINER TREE with k Steiner points can be solved in $k^{O(k)} \cdot n^{O(1)}$ time. In particular, the problem is fixed-parameter tractable.*

Theorem 1 directly extends to the ℓ_p variant of EBST for any $1 \leq p \leq \infty$, or even a broader class of metrics on \mathbb{R}^2 (such as a linear combination of ℓ_p metrics). Our algorithm combines, in a nontrivial way, previous observations for EBST, algorithmic ideas of Bae et al. [5] for the ℓ_1 variant, results from combinatorial geometry for Minkowski sums, and various new geometric insights to the problem. A key ingredient of our result is a bound on the intersection complexity of certain Minkowski sums of circular domains (Lemma 6), which might be of independent interest and can possibly find further applications.

1.1 Parameterized complexity for Steiner tree in graphs

As we consider *parameterized algorithms* for EBST, here we briefly overview the parameterized study on the Steiner tree problem in graph setting [16, 17, 21]. The classic dynamic programming algorithm for Steiner tree of Dreyfus and Wagner [18], with running time $3^{|T|} \cdot \log W \cdot n^{O(1)}$ where $|T|$ is the number of terminals, from 1971 might well be the first parameterized algorithm for *any* problem. The study of parameterized algorithms for Steiner tree has led to the design of important techniques, such as fast subset convolution [10] and the use of branching walks [35]. Research on the parameterized complexity of Steiner tree is still on-going, with very recent significant advances for the planar version of the problem [32, 36, 37]. Furthermore, algorithms for Steiner tree are frequently used as a subroutine in fixed-parameter tractable (FPT) algorithms for other problems; examples include vertex cover problems [24], near-perfect phylogenetic tree reconstruction [11], and connectivity augmentation problems [7]. Apart from $|T|$, another natural parameter associated with the problem is the number k of Steiner vertices. It is known that Steiner tree is W[2]-hard when parametrized by k [16].

2 Reducing to a fixed topology

In the first step, we reduce the original EBST problem to a variant where the “topology” of the optimal Steiner tree is given to us. This step mostly follows from the literature [5, 6], while we present it in a self-contained way. Let (P, k) be an EBST instance, where P is a set of n points in \mathbb{R}^2 and k is the parameter for the number of Steiner points. For simplicity, we call a Steiner tree on P with k Steiner points a k -BST if it is optimal in terms of bottleneck length, i.e., the maximum length of the tree edges is minimized. Our goal is just to find a k -BST on P . A *minimum spanning tree* on P , denoted by $\text{MST}(P)$, is a spanning tree with node set P that minimizes the sum of the length of its edges. For an integer $i \in [n]$, let \mathcal{F}_i be the forest obtained from $\text{MST}(P)$ by removing the $i - 1$ longest edges of $\text{MST}(P)$. Clearly, \mathcal{F}_i has i connected components. For convenience, we denote by P^2 the set of all edges connecting two points in P . Every edge of a Steiner tree on P is either in P^2 or incident to a Steiner point. We need the following property of k -BSTs.

Lemma 1 ([5, 6]). *There exists a k -BST \mathcal{T}^* on P satisfying the following.*

- For some number $K = O(k)$, \mathcal{T}^* uses all edges of \mathcal{F}_K but no other edges in P^2 .
- Every Steiner point in \mathcal{T}^* is of degree at most 5.

We can afford to guess the number K in the above lemma since $K = O(k)$. Now suppose we have K in hand and want to find a k -BST \mathcal{T}^* satisfying the conditions in Lemma 1. Let $E(\mathcal{F}_K)$ denote the set of edges of \mathcal{F}_K and C_1, \dots, C_K be the connected components of \mathcal{F}_K . With a little abuse of notations, we also use C_1, \dots, C_K to denote the corresponding sets of points in P . Thus, C_1, \dots, C_K is a partition of P . Consider the tree $\mathcal{T}^*/E(\mathcal{F}_K)$ obtained from \mathcal{T}^* by contracting all edges in $E(\mathcal{F}_K)$. Clearly, $\mathcal{T}^*/E(\mathcal{F}_K)$ contains $K + k$ nodes, among which K nodes correspond to the components C_1, \dots, C_K and the other k nodes correspond to the k Steiner points. As $K + k = O(k)$, we can afford to guess the “topology” of $\mathcal{T}^*/E(\mathcal{F}_K)$. Formally, we consider all trees of $K + k$ nodes in which K nodes are marked as C_1, \dots, C_K . The number of such trees is $k^{O(k)}$ and one of these trees, \mathcal{T}_{top} , is isomorphic to $\mathcal{T}^*/E(\mathcal{F}_K)$ where the isomorphism maps the C_i -vertex of \mathcal{T}_{top} to the C_i -vertex of $\mathcal{T}^*/E(\mathcal{F}_K)$ for every $i \in [K]$. By trying all possibilities, we can assume that \mathcal{T}_{top} is known. Note that by Lemma 1, every edge of $\mathcal{T}^*/E(\mathcal{F}_K)$ is incident to a Steiner point and every Steiner point is of degree at most 5 in $\mathcal{T}^*/E(\mathcal{F}_K)$. So we may also assume these properties of \mathcal{T}_{top} .

Let t_1, \dots, t_k be the nodes of \mathcal{T}_{top} other than the ones marked by as C_1, \dots, C_K , which correspond to the k Steiner points. For convenience, we write $S = \{t_1, \dots, t_k\}$. Now the problem becomes finding a map $\phi : S \rightarrow \mathbb{R}^2$ such that $\max_{(t,t') \in E(\mathcal{T}_{\text{top}})} d_\phi(t, t')$ is minimized, where $E(\mathcal{T}_{\text{top}})$ denotes the set of edges of \mathcal{T}_{top} . Here, $d_\phi(t, t')$ is defined as follows. If $t, t' \in S$, then $d_\phi(t, t') = \text{dist}(\phi(t), \phi(t'))$, i.e., the Euclidean distance between $\phi(t)$ and $\phi(t')$. If $t \in S$ and $t' \notin S$, then t' is marked as C_i for some $i \in [K]$ and we set $d_\phi(t, t') = \min_{x \in C_i} \text{dist}(\phi(t), x)$; the definition for the case $t \notin S$ and $t' \in S$ is symmetric. The case $t, t' \notin S$ cannot happen for any $(t, t') \in E(\mathcal{T}_{\text{top}})$, by our assumption. We call this problem *fixed-topology* EBST. In fact, we can further reduce the optimization version of fixed-topology EBST to the decision version, which aims to check whether there exists a map $\phi : S \rightarrow \mathbb{R}^2$ such that $\max_{(t,t') \in E(\mathcal{T}_{\text{top}})} d_\phi(t, t') \leq \lambda$ for a given value λ , or equivalently, $d_\phi(t, t') \leq \lambda$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$. As we demonstrate later, an FPT algorithm for the decision problem can be converted to an algorithm for the optimization problem having the same asymptotic time complexity. As demonstrated in [5], for all $i \in [K]$, given the unique point in C_i that gets connected to a Steiner point in a fixed optimal solution, the location of the optimal Steiner points can be computed in time FPT in k . One can easily find these unique points by brute-force, i.e., enumerating all possible $n^{O(k)}$ choices. As we aim for an FPT algorithm, we cannot use such a brute-force approach. Hence, we use a completely different approach than the approach of [5] in the ℓ_2 metric. However, the starting point of our approach is the FPT algorithm of [5] in the ℓ_1 metric.

3 The main algorithm

According to the discussion in Section 2, it suffices to design an algorithm for (the decision version of) fixed-topology EBST. Now we restate the setting of the problem. We have a set P of n points in \mathbb{R}^2 which is partitioned into C_1, \dots, C_K , a tree \mathcal{T}_{top} of $K + k$ nodes in which K nodes are marked as C_1, \dots, C_K (called terminal nodes) and the other k nodes are called Steiner points, and a number $\lambda > 0$. The tree \mathcal{T}_{top} satisfies (i) every edge is incident to a Steiner point and (ii) every Steiner point is of degree at most 5. With an abuse of notations, in what follows, we shall just use C_1, \dots, C_K to denote the terminal nodes of \mathcal{T}_{top} (instead of saying that they are marked as C_1, \dots, C_K). Our goal is to compute a map $\phi : S \rightarrow \mathbb{R}^2$ such that $d_\phi(t, t') \leq \lambda$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$ where $S \subseteq \mathcal{T}_{\text{top}}$ is the set of Steiner points in \mathcal{T}_{top} , or conclude the non-existence of such a map. By scaling the points in P , we can assume $\lambda = 1$ without loss of generality. Furthermore, we can assume that every leaf (i.e., a node of degree 1) of \mathcal{T}_{top} is a terminal node. Indeed, if $s \in S$ is a leaf in \mathcal{T}_{top} and $\phi_0 : S \setminus \{s\} \rightarrow \mathbb{R}^2$ is a map satisfying $d_{\phi_0}(t, t') \leq \lambda$ for all $(t, t') \in E(\mathcal{T}_{\text{top}} \setminus \{s\})$, then we can easily extend ϕ_0 to a map $\phi : S \rightarrow \mathbb{R}^2$ satisfying the desired property by choosing the point $\phi(s)$ to make $d_\phi(s, t) = 0$ where t is the only node in \mathcal{T}_{top} neighboring to s . Therefore, we can simply remove s from \mathcal{T}_{top} (and also from S). By doing this repeatedly, we can reach the situation where every leaf of \mathcal{T}_{top} is a terminal node.

In fact, we can further reduce to the situation where the leaves of \mathcal{T}_{top} are *exactly* the terminal nodes in \mathcal{T}_{top} , i.e., C_1, \dots, C_K . Consider the forest \mathcal{F} obtained from \mathcal{T}_{top} by removing the terminal nodes and the edges incident to them. The node set of \mathcal{F} is S . The connected components of \mathcal{F} induce a partition S_1, \dots, S_r of S such that there is no edge between S_i and S_j in \mathcal{T}_{top} for any $i \neq j$. Constructing a map $\phi : S \rightarrow \mathbb{R}^2$ is equivalent to constructing maps $\phi_i : S_i \rightarrow \mathbb{R}^2$ for all $i \in [r]$. Note that the construction of each map ϕ_i can be done individually, as there is no edge between S_i and S_j in \mathcal{T}_{top} for any $i \neq j$. Formally, let \mathcal{T}_i be the subtree of \mathcal{T}_{top} consisting of the nodes in S_i and their neighbors. If each ϕ_i satisfies $d_{\phi_i}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_i)$, then the map ϕ obtained by setting $\phi|_{S_i} = \phi_i$ satisfies $d_\phi(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$. Conversely, if $\phi : S \rightarrow \mathbb{R}^2$ satisfies $d_\phi(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$, then $\phi_i = \phi|_{S_i}$ must satisfy $d_{\phi_i}(t, t') \leq 1$

for all $(t, t') \in E(\mathcal{T}_i)$. Therefore, it suffices to construct each ϕ_i individually. Note that in the tree \mathcal{T}_i , every leaf is a terminal node and every internal node is a Steiner point (in S_i). With this reduction, we can assume without loss of generality that the leaves of \mathcal{T}_{top} are exactly C_1, \dots, C_K . This assumption guarantees the following nice property of the desired map ϕ , which turns out to be useful when we design our algorithm.

Lemma 2. *Assume the leaves of \mathcal{T}_{top} are exactly the terminal nodes in \mathcal{T}_{top} . If $\phi : S \rightarrow \mathbb{R}^2$ satisfies $d_\phi(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$, then $\text{dist}(\phi(a), \phi(b)) \leq k - 1$ for all $a, b \in S$.*

Proof. By the assumption, S is just the set of internal nodes of \mathcal{T}_{top} , and thus for any $a, b \in S$, the simple path π in \mathcal{T}_{top} connecting a and b only contains the nodes in S . Suppose $\pi = (t_0, t_1, \dots, t_r)$ where $t_0 = a$ and $t_r = b$. We have $r \leq k - 1$ because $|S| \leq k$. By definition, $\text{dist}(\phi(t_{i-1}), \phi(t_i)) = d_\phi(t_{i-1}, t_i) \leq 1$. Thus, by triangle inequality, $\text{dist}(\phi(a), \phi(b)) \leq \sum_{i=1}^r \text{dist}(\phi(t_{i-1}), \phi(t_i)) \leq k - 1$. \square

For convenience, we make \mathcal{T}_{top} rooted by picking an arbitrary Steiner point $\text{rt} \in S$ as the root of \mathcal{T}_{top} . Our algorithm for constructing ϕ borrows the high-level idea of *feasible regions* from the algorithm of Bae et al. [5] for the ℓ_1 variant of EBST. So before discussing our algorithm, let us first briefly review this idea. For a node $t \in \mathcal{T}_{\text{top}}$, we denote by \mathcal{T}_t the subtree of \mathcal{T}_{top} rooted at t . For each Steiner point $s \in S$, the *feasible region* $R(s) \subseteq \mathbb{R}^2$ of s is the set of all points $x \in \mathbb{R}^2$ such that there exists a map $\phi_s : S \cap \mathcal{T}_s \rightarrow \mathbb{R}^2$ satisfying $d_{\phi_s}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_s)$ and $\phi_s(s) = x$. (Bae et al. [5] defined the feasible regions in terms of the ℓ_1 metric, which is the same except that the Euclidean distance function dist is replaced with the ℓ_1 distance function.) Clearly, the desired map ϕ exists iff $R(\text{rt}) \neq \emptyset$. Furthermore, given $R(s)$ for all $s \in S$, if $R(\text{rt}) \neq \emptyset$, then one can easily obtain the map ϕ in a top-down manner as follows. Arbitrarily pick a point in $R(\text{rt})$ as $\phi(\text{rt})$. For a non-root node $s \in S$, suppose $\phi(s') \in R(s')$ is already determined for the parent s' of s and we now want to determine $\phi(s)$. Observe that there exists a point $x \in R(s)$ such that $\text{dist}(\phi(s'), x) \leq 1$. Indeed, as $\phi(s') \in R(s')$, there exists a map $\phi_{s'} : S \cap \mathcal{T}_{s'} \rightarrow \mathbb{R}^2$ satisfying $d_{\phi_{s'}}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{s'})$ and $\phi_{s'}(s') = \phi(s')$. Set $x = \phi_{s'}(s)$. By definition, $x \in R(s)$ and $\text{dist}(\phi(s'), x) \leq 1$. We then define $\phi(s) = x$. In this way, we can construct the entire map $\phi : S \rightarrow \mathbb{R}^2$ from top to bottom, which satisfies the desired property.

Bae et al. [5] showed that for the ℓ_1 case, $R(s)$ is a rectilinear domain with complexity $n^{O(1)}$ for every $s \in S$, and all these regions can be computed in $n^{O(1)}$ time in a bottom-up fashion. This directly yields a polynomial-time algorithm to solve the problem with the ℓ_1 metric. Unfortunately, in the Euclidean setting, the feasible regions are much more complicated and do not have such nice structures. As such, we cannot solve the problem by directly computing the feasible regions. Instead, our algorithm will first guess approximately the locations of the images $\phi(s)$ for $s \in S$ (as well as the locations of the points in C_1, \dots, C_K that connect to the Steiner points) and then defining feasible regions with respect to these approximate locations. In this way, we can guarantee that the feasible regions are somehow well-behaved, which finally allows us to bound their complexity by exploiting certain geometric properties of the regions as well as techniques from combinatorial geometry.

3.1 Approximately guessing the locations

Let Γ be a grid in the plane in which each cell is a closed square of side-length ε , where $\varepsilon > 0$ is a sufficiently small constant. For example, one can take $\varepsilon = 0.1$. For convenience, we also use Γ to denote the set of all cells of the grid. Our algorithm first guesses, for each Steiner point $s \in S$, which cell of Γ contains the image $\phi(s)$, and for each terminal node C_i , which cell of Γ contains the point in C_i within distance 1 from $\phi(s_i)$, where $s_i \in S$ is the parent of C_i in \mathcal{T}_{top} . At the

first glance, the number of guesses needed is at least $n^{O(k)}$, which we cannot afford. However, by applying the nice property of ϕ in Lemma 2, we can see that only $(k/\varepsilon)^{O(k)} \cdot n = k^{O(k)} \cdot n$ guesses are sufficient. Formally, we say a map $\xi : \mathcal{T}_{\text{top}} \rightarrow \Gamma$ respects a map $\phi : S \rightarrow \mathbb{R}^2$ if $\phi(s) \in \xi(s)$ for all $s \in S$ and there exists $x \in C_i \cap \xi(C_i)$ such that $\text{dist}(\phi(s_i), x) \leq 1$ for all $i \in [K]$, where $s_i \in S$ is the parent of C_i in \mathcal{T}_{top} . We have the following observation.

Lemma 3. *There exist $r = k^{O(k)} \cdot n$ maps $\xi_1, \dots, \xi_r : \mathcal{T}_{\text{top}} \rightarrow \Gamma$ such that if there exists a map $\phi : S \rightarrow \mathbb{R}^2$ satisfying $d_\phi(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$, then there also exists such a map ϕ which in addition is respected by ξ_i for some $i \in [r]$. Furthermore, the maps ξ_1, \dots, ξ_r can be computed in $O(r)$ time given P and \mathcal{T}_{top} .*

Proof. Let $\phi : S \rightarrow \mathbb{R}^2$ be a map satisfying $d_\phi(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$. Consider the terminal node $C_1 \in \mathcal{T}_{\text{top}}$. If $\xi : \mathcal{T}_{\text{top}} \rightarrow \Gamma$ respects ϕ , then the cell $\xi(C_1)$ must contain at least one point in C_1 . As $|C_1| \leq |P| = n$, there are at most n choices for $\xi(C_1)$. Suppose now $\xi(C_1) \in \Gamma$ is determined. Let o be the center of $\xi(C_1)$. In order to let ξ respect ϕ , for the parent s of C_1 in \mathcal{T}_{top} , $\phi(s)$ must be within distance $1 + \varepsilon$ from o . Furthermore, by Lemma 2, for any $s' \in S$,

$$\text{dist}(o, \phi(s')) \leq \text{dist}(o, \phi(s)) + \text{dist}(\phi(s), \phi(s')) \leq 1 + \varepsilon + (k - 1) = k + \varepsilon.$$

Therefore, the ϕ -images of the Steiner points all lie in the $O((k/\varepsilon)^2)$ cells around $\xi(C_1)$. It follows that for every Steiner point $s' \in S$, there are $O((k/\varepsilon)^2)$ choices for $\xi(s')$. Also, for every terminal node C_i , the cell $\xi(C_i)$ must be within distance $O(k)$ from $\xi(C_1)$, and thus there are $O((k/\varepsilon)^2)$ choices for $\xi(C_i)$. The total number of possibilities of ξ is then $(k/\varepsilon)^{O(K+k)} \cdot n$, which is $k^{O(k)} \cdot n$ since $1/\varepsilon = O(1)$ and $K = O(k)$. The argument above directly gives an algorithm for computing all possible maps ξ in $k^{O(k)} \cdot n$ time. \square

By computing and trying the maps ξ_1, \dots, ξ_r in the above lemma, we can guess a map $\xi : \mathcal{T}_{\text{top}} \rightarrow \Gamma$ which respects the desired map ϕ . The guess of ξ only results in an extra r factor in the running time, which is affordable as $r = k^{O(k)} \cdot n$. Our problem now becomes computing a map $\phi : S \rightarrow \mathbb{R}^2$ respected by ξ which satisfies $d_\phi(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$.

3.2 Defining feasible regions

Recall the notion of feasible regions discussed before Section 3.1. Now we slightly change the definition of feasible regions by defining them with respect to the map $\xi : \mathcal{T}_{\text{top}} \rightarrow \Gamma$. For each Steiner point $s \in S$, the *feasible region* $R(s) \subseteq \xi(s)$ of s is the set of all points $x \in \xi(s)$ such that there exists a map $\phi_s : S \cap \mathcal{T}_s \rightarrow \mathbb{R}^2$ respected by $\xi|_{\mathcal{T}_s}$ which satisfies $d_{\phi_s}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_s)$ and $\phi_s(s) = x$. For convenience, we also define the feasible region of each terminal node C_i of \mathcal{T}_{top} by setting $R(C_i) = C_i \cap \xi(C_i)$. It is clear that the desired map ϕ exists iff $R(\text{rt}) \neq \emptyset$. Furthermore, given $R(s)$ for all $s \in S$, we can easily recover the map ϕ , as discussed before Section 3.1. So it suffices to show how to compute the feasible regions efficiently. To this end, we need to first understand how the feasible regions look like.

Recall that the *Minkowski sum* of two regions A and B in \mathbb{R}^2 is defined as $A \oplus B = \{a + b : a \in A \text{ and } b \in B\}$. Let $D = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$ be the unit disk centered at the origin. For each $s \in S$, denote by $\text{Ch}(s)$ the set of children of s in \mathcal{T}_{top} . We have the following simple observation.

Lemma 4. $R(s) = (\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)) \cap \xi(s)$ for all $s \in S$.

Proof. To see $R(s) \subseteq (\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)) \cap \xi(s)$, it suffices to show $R(s) \subseteq R(s') \oplus D$ for all $s' \in \text{Ch}(s)$, because $R(s) \subseteq \xi(s)$ by definition. Let $x \in R(s)$. There exists $\phi_s : S \cap \mathcal{T}_s \rightarrow \mathbb{R}^2$

respected by $\xi|_{\mathcal{T}_s}$ which satisfies $d_{\phi_s}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_s)$ and $\phi_s(s) = x$. Consider a child $s' \in \text{Ch}(s)$. If $s' \in S$, then $\phi_s(s') \in R(s')$, because of the map $\phi_{s'}$ obtained by restricting ϕ_s to $S \cap \mathcal{T}_{s'}$. We have $\text{dist}(x, \phi_s(s')) = d_{\phi_s}(s, s') \leq 1$, which implies $x \in R(s') \oplus D$. If $s' = C_i$ for some $i \in [K]$, then there exists a point $y \in C_i \cap \xi(C_i)$ such that $\text{dist}(x, y) = \text{dist}(\phi_s(s), y) \leq 1$, which also implies $x \in R(s') \oplus D$ since $R(s') = C_i \cap \xi(C_i)$.

To see $R(s) \supseteq (\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)) \cap \xi(s)$, consider a point $x \in (\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)) \cap \xi(s)$. We construct a map $\phi_s : S \cap \mathcal{T}_s \rightarrow \mathbb{R}^2$ as follows. Set $\phi_s(s) = x$. For every $s' \in \text{Ch}(s) \cap S$, there exists a point $y \in R(s')$ such that $\text{dist}(x, y) \leq 1$. As $y \in R(s')$, there exists $\phi_{s'} : S \cap \mathcal{T}_{s'} \rightarrow \mathbb{R}^2$ respected by $\xi|_{\mathcal{T}_{s'}}$ which satisfies $d_{\phi_{s'}}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_{s'})$ and $\phi_{s'}(s') = y$. We then set $\phi_s(t) = \phi_{s'}(t)$ for all $t \in S \cap \mathcal{T}_{s'}$. It is easy to check from the construction that $\xi|_{\mathcal{T}_s}$ respects ϕ_s and $d_{\phi_s}(t, t') \leq 1$ for all $(t, t') \in E(\mathcal{T}_s)$. Thus, $x \in R(s)$. \square

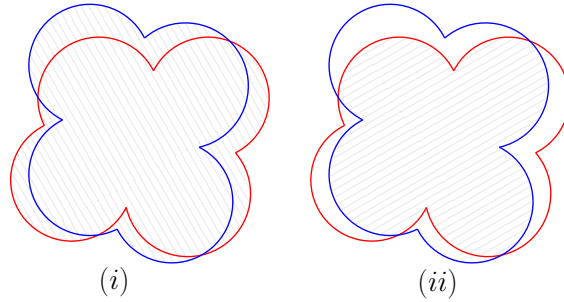


Figure 1: Illustrating (i) the union and (ii) the intersection of two pseudo-convex circular domains using shaded regions.

A *circular arc* refers to a curve that is a connected portion of a circle in \mathbb{R}^2 . A segment is also considered as a circular arc (which is a connected portion of an infinitely large circle). A *circular domain* is a closed subset (or region) of \mathbb{R}^2 whose boundary consists of circular arcs that can only intersect at their endpoints. Circular domains can be viewed as a generalization of polygons whose boundaries consist of segments and are not self-intersecting. *For our convenience, by abusing notation, we also allow any finite set of points to be a circular domain.* The *complexity* of any given circular domain R , denoted by $\|R\|$, is defined as follows depending on the structure of R . If R is not just a set of points, $\|R\|$ is the total number of circular arcs and vertices (i.e., the intersection points of the arcs) on its boundary. Otherwise, if R is a set of points, $\|R\|$ is just the number of these points.

A circular domain R is called *pseudo-convex* if for every boundary arc σ of R that is not a segment, the side of σ corresponding to the interior of R coincides with the side of σ corresponding to the interior of the disk defining σ . Clearly, the union and the intersection of pseudo-convex circular domains are still pseudo-convex circular domains (see Figure 1 for examples). Furthermore, one can verify that if R is a pseudo-convex circular domain, then the Minkowski sum $R \oplus D$ is also a pseudo-convex circular domain (see Figure 2 for examples). Therefore, by Lemma 4, one can see that $R(s)$ is a pseudo-convex circular domain for every $s \in S$. For each terminal node C_i , $R(C_i)$ is a set of discrete points, which we also view as a pseudo-convex circular domain (with only vertices and no arcs) for convenience. Using the formula in Lemma 4 for $R(s)$, one can then compute $R(s)$ in time polynomial in $\sum_{s' \in \text{Ch}(s)} \|R(s')\|$. Indeed, each Minkowski sum $R(s') \oplus D$ can be obtained by computing the Minkowski sum of every boundary arc of $R(s')$ and D , while the intersection of circular domains can be computed by constructing the arrangement of their boundary arcs. As the

center of the circle corresponding to any such arc can be computed in an inductive manner, each arc can be expressed using $O(1)$ complexity. Hence, the intersection points of two such arcs can also be computed in constant time. Next, we bound the complexity of the feasible regions.

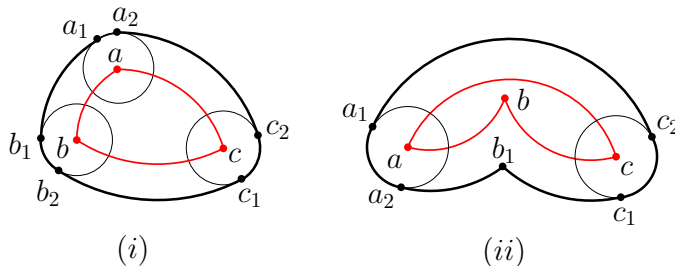


Figure 2: Illustrating the Minkowski sums of pseudo-convex circular domain $\widehat{ab}, \widehat{bc}, \widehat{ca}$ and the unit disk D . (i) The sum is shown by $\widehat{a_1b_1}, \widehat{b_1b_2}, \widehat{b_2c_1}, \widehat{c_1c_2}, \widehat{c_2a_2}$, and $\widehat{a_2a_1}$. (ii) The sum is shown by $\widehat{a_1a_2}, \widehat{a_2b_1}, \widehat{b_1c_1}, \widehat{c_1c_2}$, and $\widehat{c_2a_1}$.

3.3 The complexity of feasible regions

In this section, we bound $\|R(s)\|$ for the Steiner points $s \in S$. Define the *level* $L(t)$ of a node $t \in \mathcal{T}_{\text{top}}$ by setting $L(t) = 0$ if t is a leaf and $L(t) = \max_{t' \in \text{Ch}(t)} L(t') + 1$ otherwise. Clearly, $L(s) \geq 1$ for all $s \in S$ and $L(C_i) = 0$ for all $i \in [K]$. As $|S| \leq k$, the maximum level of a node in \mathcal{T}_{top} is bounded by k . By Lemma 4, the feasible region of $s \in S$ depends on the feasible regions of its children whose levels are strictly smaller than $L(s)$. Thus, it is natural to study how the complexity of feasible regions increases along with the level of the nodes.

The nodes of \mathcal{T}_{top} at level 0 are exactly C_1, \dots, C_K . We have $\|R(C_i)\| = O(n)$ for every $i \in [K]$, as $R(C_i) = C_i \cap \xi(C_i) \subseteq P$ consists of at most n points in the plane. Assume $\|R(t)\| \leq m_i$ for all nodes $t \in \mathcal{T}_{\text{top}}$ with $L(t) \leq i$. Consider a node $s \in S$ with $L(s) = i + 1$. Now $\|R(s')\| \leq m_i$ for all $s' \in \text{Ch}(s)$, and we want to bound $\|R(s)\|$. As $R(s) = (\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)) \cap \xi(s)$ by Lemma 4, the first step is to bound $\|R(s') \oplus D\|$ for $s' \in \text{Ch}(s)$. The complexity of Minkowski sums of polygons is well-studied. Surprisingly, there is not much work in the literature studying Minkowski sums of circular domains. However, it is not surprising that the insights for Minkowski sums of polygons can also be applied to circular domains. We show in the following lemma that the known arguments for polygons together with the standard technique of vertical decomposition result in, at least, a linear bound on the complexity of the Minkowski sum of a pseudo-convex circular domain and a unit disk (which is sufficient for our purpose).

Lemma 5. *For any pseudo-convex circular domain R , we have $\|R \oplus D\| = O(\|R\|)$.*

Proof. If R is convex, then one can directly apply the argument for bounding the complexity of the Minkowski sum of two convex polygons (see for example [8]) to show that $\|R \oplus D\| = O(\|R\| + \|D\|)$, which is just $O(\|R\|)$. In fact, in this case, one can even show that each vertex/arc of R contributes at most one arc of $R \oplus D$, and thus $\|R \oplus D\| \leq 2\|R\|$.

When R is not convex (but pseudo-convex), we shall apply the known argument for bounding the complexity of the Minkowski sum of a non-convex polygon and a convex polygon [29]. Consider the (standard) vertical decomposition of R . Specifically, for each vertex v on the boundary of R , we shoot two rays originated from v directing upward and downward, respectively; the rays stop when they touch the boundary of R . Those rays cut R into smaller circular domains R_1, \dots, R_r with total

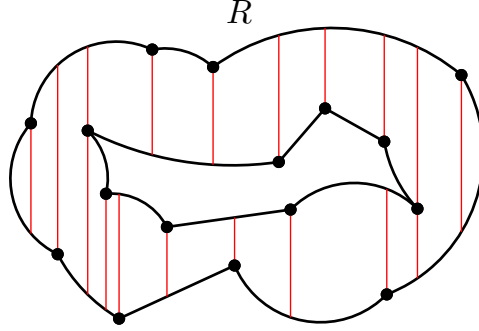


Figure 3: Illustrating the vertical decomposition of R .

complexity $O(\|R\|)$. See Figure 3 for an illustration of this vertical decomposition of R . It is easy to observe that each of these smaller circular domains is convex. Indeed, each R_i is pseudo-convex, as R is pseudo-convex. Furthermore, the angle of R_i at every vertex v is at most π (here the angle is defined by the two arcs of R_i incident to v with the side corresponding to the interior of R_i), for otherwise the angle is cut by one of the two rays originated from v and thus cannot survive in R_i . Therefore, R_i is convex. It was shown in [29] that if P and Q are two interior-disjoint convex regions (in \mathbb{R}^2), and Z is another convex region, then $P \oplus Z$ and $Q \oplus Z$ are pseudo-disks, i.e., their boundary cross each other at most twice. Now R_1, \dots, R_r are interior-disjoint and convex, which implies $R_1 \oplus D, \dots, R_r \oplus D$ are pseudo-disks. Note that $R \oplus D = \bigcup_{i=1}^r (R_i \oplus D)$. By the linear union complexity of pseudo-disks [29], we have $\|\bigcup_{i=1}^r (R_i \oplus D)\| = O(\sum_{i=1}^r \|R_i \oplus D\|)$, and the latter is $O(\sum_{i=1}^r \|R_i\|)$ because each R_i is convex. Finally, as the total complexity of R_1, \dots, R_r is $O(\|R\|)$, we have $\|R \oplus D\| = O(\|R\|)$. \square

Applying the above lemma, we have $\|R(s') \oplus D\| = O(m_i)$ for all $s' \in \text{Ch}(s)$. Next, we are going to bound the complexity of the intersection $\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)$. This step is more challenging, and is achieved by proving the following key lemma.

Lemma 6. *Let R_1, \dots, R_r be circular domains each of which is inside an $\varepsilon \times \varepsilon$ grid cell for $\varepsilon \leq 0.1$. Then $\|\bigcap_{i=1}^r (R_i \oplus D)\| = O(r^2 M)$, where $M = \max_{i=1}^r \|R_i \oplus D\|$.*

As the proof of Lemma 6 is technical, we defer it to Section 3.3.1. At this point, let us first finish the discussion assuming this lemma. By Lemma 6, we have $\|\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)\| = O(r^2 m_i)$, where $r = |\text{Ch}(s)|$. Recall that each Steiner point in \mathcal{T}_{top} is of degree at most 5, which implies $r \leq 5$. Therefore, $\|\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)\| = O(m_i)$. To further bound $\|R(s)\|$ is easy. By Lemma 4, $R(s)$ is the intersection of $\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)$ and the grid cell $\xi(s)$. Note that taking intersection with a square can only increase the complexity by a constant factor. Indeed, $\xi(s)$ has four edges and each edge intersects each boundary arc of $\bigcap_{s' \in \text{Ch}(s)} (R(s') \oplus D)$ at most twice. Therefore, $\|R(s)\| = O(m_i)$. In other words, from level i to level $i + 1$, the complexity of the feasible regions only increase by a constant factor. It follows that $\|R(s)\| = 2^{O(j)} \cdot n$ for any node $s \in \mathcal{T}_{\text{top}}$ with $L(s) = j$. Since \mathcal{T}_{top} has at most k levels, we conclude that $\|R(s)\| = 2^{O(k)} \cdot n$ for all $s \in \mathcal{T}_{\text{top}}$.

3.3.1 Proof of Lemma 6

In this section, we prove Lemma 6. Let R_1, \dots, R_r be the circular domains in the lemma. We prove that each $R_i \oplus D$ is a *simple* circular domain (i.e., connected and without holes). In fact, we prove an even stronger property of $R_i \oplus D$.

Observation 1. Let \square be an $\varepsilon \times \varepsilon$ grid cell and $R \subseteq \square$ be any set. For any points $o \in \square$ and $p \in R \oplus D$, the segment \overline{op} is contained in $R \oplus D$, and furthermore the interior of \overline{op} is contained in the interior of $R \oplus D$.

Proof. Since $p \in R \oplus D$, there exists $q \in R \subseteq \square$ such that $p \in D_q$, where D_q is the unit disk centered at q . As $o, q \in \square$ and the side-length of \square is $\varepsilon \leq 0.1$ by the assumption in Lemma 6, we have $o \in D_q$. By the convexity of D_q , we have $\overline{op} \subseteq D_q$. Note that $D_q \subseteq R \oplus D$ because $q \in R$, which implies $\overline{op} \subseteq R \oplus D$. Furthermore, the interior of \overline{op} is in the interior of D_q and thus in the interior of $R \oplus D$. \square

The above observation implies that each $R_i \oplus D$ is star-shaped (i.e., there exists one point $o \in R_i \oplus D$ that sees the entire region $R_i \oplus D$), and in particular, a simple circular domain. We use $\partial(R_i \oplus D)$ to denote the boundary of $R_i \oplus D$, which is a simple closed circular curve (i.e., curve consisting of circular arcs). Consider the arrangement \mathcal{A} of all circular curves $\partial(R_1 \oplus D), \dots, \partial(R_r \oplus D)$. The vertices of \mathcal{A} are the vertices of the curves $\partial(R_1 \oplus D), \dots, \partial(R_r \oplus D)$ and their proper intersection points¹. These vertices subdivide the curves $\partial(R_1 \oplus D), \dots, \partial(R_r \oplus D)$ into smaller pieces each of which is a circular arc; they are the edges of \mathcal{A} . Finally, the curves $\partial(R_1 \oplus D), \dots, \partial(R_r \oplus D)$ subdivide the plane \mathbb{R}^2 into connected regions, which are the faces of \mathcal{A} . Clearly, $\bigcap_{i=1}^r (R_i \oplus D)$ is the union of several faces of \mathcal{A} . Therefore, $\|\bigcap_{i=1}^r (R_i \oplus D)\|$ is bounded by the total number of vertices and edges of \mathcal{A} . We first consider the number of vertices of \mathcal{A} (bounding the number of edges is easy once we know the number of vertices).

The total number of vertices of the curves $\partial(R_1 \oplus D), \dots, \partial(R_r \oplus D)$ is bounded by $O(rM)$ where $M = \max_{i=1}^r \|R_i \oplus D\|$. Besides these points, the other vertices of \mathcal{A} are proper intersection points of $\partial(R_i \oplus D)$ and $\partial(R_j \oplus D)$. In what follows, we shall prove that the number of proper intersection points of $\partial(R_i \oplus D)$ and $\partial(R_j \oplus D)$ is bounded by $O(M)$, for any $i, j \in [r]$. Note that this bound does not hold for *general* circular curves. Indeed, one can easily show that two circular curves of complexity M can have $\Theta(M^2)$ proper intersection points in worst case (even if both curves are boundaries of star-shaped pseudo-convex circular domains). Our proof heavily relies on the fact that the two curves are both the boundaries of the Minkowski sums of a “tiny” circular domain and the unit disk D . More specifically, the geometry of such Minkowski sums allows us to somehow relate the two curves with *monotone* curves, which are well-behaved. Without loss of generality, it suffices to consider the curves $\partial(R_1 \oplus D)$ and $\partial(R_2 \oplus D)$, i.e., the case $i = 1$ and $j = 2$.

Recall that a curve γ in \mathbb{R}^2 is *x-monotone* if there exists a homeomorphism $f : [0, 1] \rightarrow \gamma$ such that the x -coordinate of $f(a)$ is smaller than the x -coordinate of $f(b)$ whenever $a < b$. Intuitively, γ is *x-monotone* if we always move right (or left) when going along γ from one side to the other side. By slightly generalizing this notion, we can define the monotonicity of a curve with respect to a vector. Formally, let $\vec{v} \in \mathbb{S}^1$ be a unit vector. We say a curve γ in \mathbb{R}^2 is *\vec{v} -monotone* if there exists a homeomorphism $f : [0, 1] \rightarrow \gamma$ such that $\langle \vec{v}, f(a) \rangle < \langle \vec{v}, f(b) \rangle$ whenever $a < b$; here $\langle \cdot, \cdot \rangle$ denotes the inner product. Two curves γ and γ' in \mathbb{R}^2 are *mutually monotone* if there exists $\vec{v} \in \mathbb{S}^1$ such that γ and γ' are both \vec{v} -monotone. One can easily show that two mutually monotone curves have linear number of proper intersection points.

Fact 7. Let γ and γ' be two circular curves that are mutually monotone. Then γ and γ' have $O(m + m')$ proper intersection points, where $m = \|\gamma\|$ and $m' = \|\gamma'\|$.

¹A point $p \in \mathbb{R}^2$ is a *proper intersection point* of two curves γ and γ' in \mathbb{R}^2 , if $p \in \gamma \cap \gamma'$ and p is *isolated* in $\gamma \cap \gamma'$, i.e., there is an open neighborhood $U \subseteq \mathbb{R}^2$ of p such that $U \cap (\gamma \cap \gamma') = \{p\}$. It can happen that two curves $\partial(R_i \oplus D)$ and $\partial(R_j \oplus D)$ have infinitely many intersection points when an arc of $\partial(R_i \oplus D)$ overlaps with an arc of $\partial(R_j \oplus D)$. But these are not proper intersection points and do not contribute vertices of \mathcal{A} .

Proof. Without loss of generality, assume γ and γ' are x -monotone. Let V (resp., V') denote the set of vertices on γ (resp., γ'). Suppose the x -coordinates of the points in $V \cup V'$ are x_0, x_1, \dots, x_r , where $r = O(m + m')$ and $x_0 \leq x_1 \leq \dots \leq x_r$. For each $i \in [r]$, define $X_i = [x_{i-1}, x_i] \times \mathbb{R}^2$, which is a vertical strip. The part of γ (resp., γ') inside each X_i is a circular arc. Two circular arcs can have at most two proper intersection points. As such, γ and γ' have at most two proper intersection points inside each X_i , and thus have in total at most $2r = O(m + m')$ proper intersection points. \square

Based on the above observation, our key idea is to decompose $\partial(R_1 \oplus D)$ and $\partial(R_2 \oplus D)$ into $O(1)$ pieces such that any two of these pieces are mutually monotone. As long as this is possible, we can easily bound the number of proper intersection points of $\partial(R_1 \oplus D)$ and $\partial(R_2 \oplus D)$. To do this decomposition, we need to first establish some nice geometric properties of the Minkowski sum of a “tiny” circular domain and the unit disk D .

Fix a circular domain R that is inside an $\varepsilon \times \varepsilon$ grid cell \square , and let o be the center of \square . Observation 1 implies that any ray r originated from o intersects $\partial(R \oplus D)$ at exactly one point. Indeed, if r intersects $\partial(R \oplus D)$ at two points a and b where a is closer to o than b , then the interior of \overline{ob} is not contained in the interior of $R \oplus D$ as (a is in the interior of \overline{ob} but not in the interior of $R \oplus D$), which contradicts Observation 1. It follows that the map $\pi : \partial(R \oplus D) \rightarrow \mathbb{S}^1$ defined as $\pi(p) = \overrightarrow{op} / \|\overrightarrow{op}\|$ is bijective and is thus a homeomorphism. Therefore, $\partial(R \oplus D)$ has the nice property that if a point p moves along $\partial(R \oplus D)$ in one direction, then the vector \overrightarrow{op} also rotates in one direction.

A point $p \in \partial(R \oplus D)$ is a *non-vertex point* if it is in the interior of a circular arc of $\partial(R \oplus D)$. Consider a non-vertex point $p \in \partial(R \oplus D)$ lying in the interior of the circular arc σ of $\partial(R \oplus D)$. There exists a unique line ℓ going through p that is tangent to $\partial(R \oplus D)$, which is also the tangent line of σ at p . A *tangent vector* of $R \oplus D$ at p refers to a unit vector parallel to ℓ . Clearly, there are two tangent vectors of $R \oplus D$ at p , among which one vector \vec{v} indicates the clockwise direction in the sense that if p moves along $\partial(R \oplus D)$ in the direction of \vec{v} , then \overrightarrow{op} rotates clockwise. We then call \vec{v} the *clockwise tangent vector* (or *clockwise tangent* for short) of $R \oplus D$ at p , and denote it by \tan_p . In the next observation, we prove that \tan_p is almost perpendicular to the vector \overrightarrow{op} . For two nonzero vectors \vec{u} and \vec{v} in the plane, let $\text{ang}(\vec{u}, \vec{v})$ denote the *clockwise ordered angle* from \vec{u} to \vec{v} , i.e., the angle between \vec{u} and \vec{v} that is to the clockwise of \vec{u} and to the counter-clockwise of \vec{v} .

Observation 2. *For any non-vertex point $p \in \partial(R \oplus D)$, we have $|\text{ang}(\overrightarrow{op}, \tan_p) - \frac{\pi}{2}| \leq 4\varepsilon$, where \tan_p is the clockwise tangent of $R \oplus D$ at p .*

Proof. Consider a non-vertex point $p \in \partial(R \oplus D)$ and suppose it is in the interior of a circular arc σ of $\partial(R \oplus D)$. As $p \in \partial(R \oplus D)$, there exists a point $q \in R$ such that $\text{dist}(p, q) = 1$. Let D_q be the unit disk centered at q , whose boundary ∂D_q contains p . Now the circular arc σ and the circle ∂D_q intersect at p . But σ and ∂D_q cannot cross each other at p , because σ is a portion of $\partial(R \oplus D)$ and $D_q \subseteq R \oplus D$. Therefore, σ and ∂D_q are tangent at p , and they share the same tangent line ℓ at p . See Figure 4 for an illustration. As a tangent line of D_q at p , ℓ is perpendicular to \overrightarrow{qp} . We claim that the (smaller) angle between ℓ and \overrightarrow{op} is at least $\frac{\pi}{2} - 4\varepsilon$. It suffices to show that $\angle opq \leq 4\varepsilon$, since the angle between ℓ and \overrightarrow{qp} is $\frac{\pi}{2}$. We have $\text{dist}(p, q) = 1$ and $\text{dist}(o, q) \leq 2\varepsilon$. By the formula $\frac{\text{dist}(o, q)}{\text{dist}(p, q)} = \frac{\sin \angle opq}{\sin \angle poq}$, we have that $\frac{\sin \angle opq}{\sin \angle poq} \leq 2\varepsilon$ and thus $\sin \angle opq \leq 2\varepsilon$. It follows that $\angle opq \leq 4\varepsilon$. Let \tan_p be the clockwise tangent of $R \oplus D$ at p . Since the angle between ℓ and \overrightarrow{op} is at least $\frac{\pi}{2} - 4\varepsilon$, we have either $|\text{ang}(\overrightarrow{op}, \tan_p) - \frac{\pi}{2}| \leq 4\varepsilon$ or $|\text{ang}(\overrightarrow{op}, \tan_p) - \frac{3\pi}{2}| \leq 4\varepsilon$. But by the definition of *clockwise tangent*, $\text{ang}(\overrightarrow{op}, \tan_p) \leq \pi$. Therefore, $|\text{ang}(\overrightarrow{op}, \tan_p) - \frac{\pi}{2}| \leq 4\varepsilon$. \square

The next observation allows us to test the monotonicity of a piece of $\partial(R \oplus D)$ by checking the clockwise tangent vectors at points on that piece.

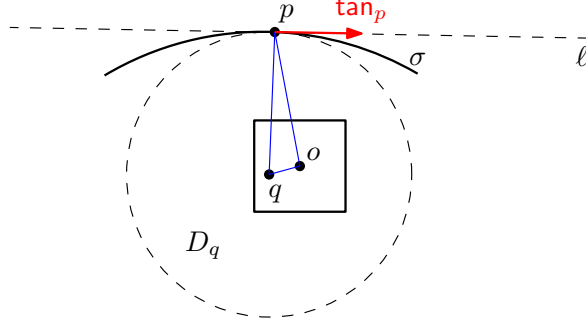


Figure 4: Illustrating the proof of Observation 2.

Observation 3. Let γ be a connected portion of $\partial(R \oplus D)$ and $\vec{v} \in \mathbb{S}^1$ be a unit vector. If $\langle \vec{v}, \tan_p \rangle > 0$ for any non-vertex point $p \in \gamma$ or $\langle \vec{v}, \tan_p \rangle < 0$ for any non-vertex point $p \in \gamma$, then γ is \vec{v} -monotone.

Proof. Without loss of generality, we only need to consider the case $\langle \vec{v}, \tan_p \rangle > 0$ for any non-vertex point $p \in \gamma$. Let $a, b \in \gamma$ be the two endpoints of γ such that γ is on the clockwise (resp., counterclockwise) side of a (resp., b). Consider a homeomorphism $f : [0, 1] \rightarrow \gamma$ with $f(0) = a$ and $f(1) = b$. Define $I = \{x \in [0, 1] : f(x) \text{ is a vertex point}\}$ and suppose $I = \{x_1, \dots, x_{r-1}\}$ where $x_1 < \dots < x_{r-1}$. Set $x_0 = 0$ and $x_r = 1$. For each $i \in [r]$, the image $f([x_{i-1}, x_i])$ is a circular arc on $\partial(R \oplus D)$. When x goes from x_{i-1} to x_i , $f(x)$ is moving clockwise around o . As $\langle \vec{v}, \tan_{f(x)} \rangle > 0$ for any $x \in (x_{i-1}, x_i)$ and $f(x)$ is moving clockwise around o when x goes from x_{i-1} to x_i , the function $g(x) = \langle \vec{v}, f(x) \rangle$ is increasing on the open interval (x_{i-1}, x_i) and is thus increasing on the closed interval $[x_{i-1}, x_i]$ because it is continuous. Since $[0, 1] = \bigcup_{i=1}^r [x_{i-1}, x_i]$, $g(x)$ is increasing on the entire range $[0, 1]$, which implies that γ is \vec{v} -monotone. \square

Let $\Lambda \geq 10$ be an integer. Consider a subdivision of $\partial(R \oplus D)$ into Λ pieces as follows. We shoot Λ rays from o , which evenly divide the 2π angle around o into Λ angles each of size $2\pi/\Lambda$. These rays intersect $\partial(R \oplus D)$ at Λ points p_1, \dots, p_Λ such that $\text{ang}(\vec{op}_{i-1}, \vec{op}_i) = 2\pi/\Lambda$; for convenience, here we set $p_0 = p_\Lambda$. The points p_1, \dots, p_Λ subdivide $\partial(R \oplus D)$ into Λ circular curves $\gamma_1, \dots, \gamma_\Lambda$, where γ_i is the one with endpoints p_{i-1} and p_i . We call $\gamma_1, \dots, \gamma_\Lambda$ a Λ -decomposition of $\partial(R \oplus D)$. See Figure 5 for an illustration (while we require $\Lambda \geq 10$, the figure only shows $\Lambda = 8$ for simplicity). We observe that each curve γ_i is monotone with respect to almost all vectors $\vec{v} \in \mathbb{S}^1$ (if Λ is sufficiently large and ε is sufficiently small).

Observation 4. For every $i \in [\Lambda]$, γ_i is \vec{v} -monotone for all $\vec{v} \in \mathbb{S}^1$ such that $\varepsilon' \leq \text{ang}(\vec{op}_i, \vec{v}) \leq \pi - \varepsilon'$ or $\pi + \varepsilon' \leq \text{ang}(p_i, \vec{v}) \leq 2\pi - \varepsilon'$, where $\varepsilon' = 7(\frac{1}{\Lambda} + \varepsilon)$.

Proof. Let $\vec{v} \in \mathbb{S}^1$. Without loss of generality, we only need to consider the case $\varepsilon' \leq \text{ang}(\vec{op}_i, \vec{v}) \leq \pi - \varepsilon'$ because γ_i is \vec{v} -monotone iff γ_i is $(-\vec{v})$ -monotone. By construction, we have $\text{ang}(\vec{op}, \vec{op}_i) \leq 2\pi/\Lambda$ for any non-vertex point $p \in \gamma_i$. Furthermore, by Observation 2, we have $|\text{ang}(\vec{op}, \tan_p) - \frac{\pi}{2}| \leq 4\varepsilon$ for any non-vertex point $p \in \gamma_i$. As $\Lambda \geq 10$ and $\varepsilon \leq 0.1$, both $2\pi/\Lambda$ and 4ε are sufficiently small. So we have $|\text{ang}(\vec{op}_i, \tan_p) - \frac{\pi}{2}| \leq 2\pi/\Lambda + 4\varepsilon$ for any non-vertex point $p \in \gamma_i$. We claim that $\langle \vec{v}, \tan_p \rangle > 0$ for any non-vertex point $p \in \gamma_i$. Let $\vec{u} \in \mathbb{S}^1$ be the unique vector satisfying $\text{ang}(\vec{op}_i, \vec{u}) = \frac{\pi}{2}$. Then the smaller angle between \vec{v} and \vec{u} is at most $\frac{\pi}{2} - \varepsilon'$. By the fact $|\text{ang}(\vec{op}_i, \tan_p) - \frac{\pi}{2}| \leq 2\pi/\Lambda + 4\varepsilon$, we know that the smaller angle between \tan_p and \vec{u} is at most $2\pi/\Lambda + 4\varepsilon$, which is strictly smaller than $7(\frac{1}{\Lambda} + \varepsilon) = \varepsilon'$. Therefore, the smaller angle between \vec{v} and \tan_p is strictly smaller than $\frac{\pi}{2}$, which implies $\langle \vec{v}, \tan_p \rangle > 0$. Finally, by Observation 3, γ_i is \vec{v} -monotone. \square

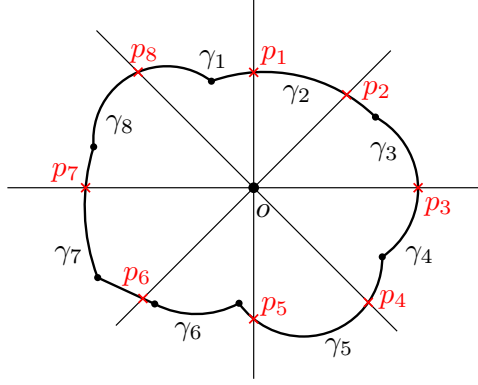


Figure 5: Illustrating the Λ -decomposition for $\Lambda = 8$.

Now we are ready to bound the number of proper intersection points of $\partial(R_1 \oplus D)$ and $\partial(R_2 \oplus D)$. Take a Λ -decomposition $\gamma_1, \dots, \gamma_\Lambda$ of $\partial(R_1 \oplus D)$ and a Λ -decomposition $\eta_1, \dots, \eta_\Lambda$ of $\partial(R_2 \oplus D)$ for $\Lambda = 10$. Observe that γ_i and η_j are mutually monotone for all $i, j \in [\Lambda]$. To see this, let $V_\gamma = \{\vec{v} \in \mathbb{S}^1 : \gamma_i \text{ is } \vec{v}\text{-monotone}\}$ and $V_\eta = \{\vec{v} \in \mathbb{S}^1 : \eta_j \text{ is } \vec{v}\text{-monotone}\}$. By Observation 4, V_γ contains two disjoint arcs on \mathbb{S}^1 each of length $\pi - 2\varepsilon'$, and so does V_η . As $\Lambda = 10$ and $\varepsilon \leq 0.1$, we have $2\varepsilon' \leq 1.4 \leq \frac{\pi}{2}$ and thus $\pi - 2\varepsilon' \geq \frac{\pi}{2}$. Therefore, V_γ (resp., V_η) contains two disjoint arcs on \mathbb{S}^1 each of length at least $\frac{\pi}{2}$. It follows that $V_\gamma \cap V_\eta \neq \emptyset$, which further implies that γ_i and η_j are mutually monotone, as both of them are \vec{v} -monotone for any $\vec{v} \in V_\gamma \cap V_\eta$. By Fact 7, the number proper intersection points of γ_i and η_j is $O(M)$. As $\Lambda = 10$, we know that $\partial(R_1 \oplus D)$ and $\partial(R_2 \oplus D)$ have $O(M)$ proper intersection points.

As the number of proper intersection points of any two curves $\partial(R_i \oplus D)$ and $\partial(R_j \oplus D)$ is $O(M)$, the total number of vertices of the arrangement \mathcal{A} is $O(r^2M)$. By the definition of \mathcal{A} , it may contain overlapping edges. Thus, it is not exactly a planar graph. However, it is not hard to see that the number of edges of \mathcal{A} is $O(r^3M)$. Indeed, the number of edges of \mathcal{A} is at most $2r$ times the number of vertices of \mathcal{A} , as each vertex can be incident to at most $2r$ edges (each of the r curves contributes at most two edges). In fact, we can improve the bound to $O(r^2M)$ as follows. For each vertex v of \mathcal{A} , define $\Delta_v = \Delta'_v + \Delta''_v$, where Δ'_v is the number of indices $i \in [r]$ such that v is a vertex of $\partial(R_i \oplus D)$ and Δ''_v is the number of pairs $(i, j) \in [r] \times [r]$ such that v is a proper intersection point of $\partial(R_i \oplus D)$ and $\partial(R_j \oplus D)$. We claim that $\sum_v \Delta'_v = O(rM)$ and $\sum_v \Delta''_v = O(r^2M)$. Indeed, each vertex of each curve $\partial(R_i \oplus D)$ is counted once in $\sum_v \Delta'_v$ and each proper intersection point of each pair of curves is counted once in $\sum_v \Delta''_v$. As each curve $\partial(R_i \oplus D)$ has $O(M)$ vertices and each pair of curves have $O(M)$ proper intersection points as shown before, we have $\sum_v \Delta'_v = O(rM)$ and $\sum_v \Delta''_v = O(r^2M)$, which implies $\sum_v \Delta_v = O(r^2M)$. Clearly, the number of edges of \mathcal{A} incident to each vertex v is bounded by $O(\Delta_v)$. Therefore, the total number of edges of \mathcal{A} is $O(r^2M)$. Finally, we conclude that $\|\bigcap_{i=1}^r (R_i \oplus D)\| = O(r^2M)$.

3.4 Putting everything together

In Section 3.3, we obtain the bound $\|R(s)\| = 2^{O(k)} \cdot n$ for all $s \in \mathcal{T}_{\text{top}}$. Combining it with the discussion in Section 3.2, we can compute a map $\phi : S \rightarrow \mathbb{R}^2$ respected by a given $\xi : \mathcal{T}_{\text{top}} \rightarrow \Gamma$ satisfying $d_\phi(t, t')$ for all $(t, t') \in E(\mathcal{T}_{\text{top}})$ or decide the non-existence of such a map in $2^{O(k)} \cdot n^{O(1)}$ time. Further combining this with our guess for ξ in Section 3.1, we obtain a $k^{O(k)}n^{O(1)}$ -time algorithm for the decision version of fixed-topology EBST.

Recall the standard parametric search technique of Megiddo [33], which can convert a decision

algorithm \mathcal{A} for some problem with running time $T(n)$ to an optimization algorithm for the same problem with running time $O(T^2(n))$, by using \mathcal{A} as both the test algorithm and the decision algorithm. Therefore, our $k^{O(k)}n^{O(1)}$ -time decision algorithm implies a $k^{O(k)}n^{O(1)}$ -time optimization algorithm (for fixed-topology EBST) as well. Finally, combining this with the reduction in Section 2, we obtain a $k^{O(k)}n^{O(1)}$ -time algorithm for EBST.

Theorem 1. EUCLIDEAN BOTTLENECK STEINER TREE with k Steiner points can be solved in $k^{O(k)} \cdot n^{O(1)}$ time. In particular, the problem is fixed-parameter tractable.

One can verify that our proof of Theorem 1 also works for the ℓ_p variant of EBST for any rational $1 \leq p \leq \infty$. In the ℓ_p setting, everything is the same except that the unit disk D becomes an ℓ_p unit disk. Consequently, the boundary arcs of the “circular” domains become ℓ_p arcs. However, our arguments did not use any special property of *circular* arcs, and thus still apply (possibly with different parameters ε and Λ). More generally, our algorithm works as long as D is convex and its boundary consists of $O(1)$ algebraic curves. Due to the $O(1)$ complexity of these curves, the intersection points of any two of them can be computed in constant time. As such, Theorem 1 can also be extended to some other metrics on \mathbb{R}^2 , such as a positive linear combination of ℓ_p metrics.

References

- [1] A Karim Abu-Affash. On the euclidean bottleneck full steiner tree problem. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 433–439, 2011. 2
- [2] A Karim Abu-Affash, Sujoy Bhore, Paz Carmi, and Dibyayan Chakraborty. Bottleneck bichromatic full steiner trees. *Information Processing Letters*, 142:14–19, 2019. 2
- [3] A Karim Abu-Affash, Paz Carmi, Matthew J Katz, and Michael Segal. The euclidean bottleneck steiner path problem. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 440–447, 2011. 2
- [4] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998. 1
- [5] Sang Won Bae, Sunghee Choi, Chunseok Lee, and Shin-ichi Tanigawa. Exact algorithms for the bottleneck steiner tree problem. *Algorithmica*, 61(4):924–948, 2011. 2, 3, 4, 5
- [6] Sang Won Bae, Chunseok Lee, and Sunghee Choi. On exact solutions to the euclidean bottleneck steiner tree problem. *Information Processing Letters*, 110(16):672–678, 2010. 2, 3
- [7] Manu Basavaraju, Fedor V. Fomin, Petr A. Golovach, Pranabendu Misra, M. S. Ramanujan, and Saket Saurabh. Parameterized algorithms to preserve connectivity. In *Proceedings of the 41st International Colloquium of Automata, Languages and Programming (ICALP)*, volume 8572 of *Lecture Notes in Comput. Sci.*, pages 800–811. Springer, 2014. 3
- [8] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. Computational geometry. In *Computational geometry*, pages 1–17. Springer, 1997. 8
- [9] Ahmad Biniiaz, Anil Maheshwari, and Michiel Smid. An optimal algorithm for the euclidean bottleneck full steiner tree problem. *Computational Geometry*, 47(3):377–380, 2014. 2

- [10] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–74, New York, 2007. ACM. 1, 3
- [11] Guy E. Blleloch, Kedar Dhamdhare, Eran Halperin, R. Ravi, Russell Schwartz, and Srinath Sridhar. Fixed parameter tractability of binary near-perfect phylogenetic tree reconstruction. In *Proceedings of the 33rd International Colloquium of Automata, Languages and Programming (ICALP)*, volume 4051 of *Lecture Notes in Comput. Sci.*, pages 667–678. Springer, 2006. 3
- [12] Marcus Brazil, Ronald L Graham, Doreen A Thomas, and Martin Zachariasen. On the history of the euclidean steiner tree problem. *Archive for history of exact sciences*, 68(3):327–354, 2014. 1
- [13] Marcus Brazil, Charl J Ras, Konrad J Swanepoel, and Doreen A Thomas. Generalised k-steiner tree problems in normed planes. *Algorithmica*, 71(1):66–86, 2015. 2
- [14] Donghui Chen, Ding-Zhu Du, Xiao-Dong Hu, Guo-Hui Lin, Lusheng Wang, and Guoliang Xue. Approximations for steiner trees with minimum number of steiner points. *Theoretical Computer Science*, 262(1-2):83–99, 2001. 2
- [15] Charles C Chiang, Majid Sarrafzadeh, and Chak-Kuen Wong. A powerful global router: based on steiner min-max trees. In *ICCAD*, pages 2–5. Citeseer, 1989. 1, 2
- [16] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3. 1, 3
- [17] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. 1, 3
- [18] Stuart E. Dreyfus and Robert A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971. 1, 3
- [19] Dingzhu Du and Xiaodong Hu. *Steiner tree problems in computer communication networks*. World Scientific, 2008. 1
- [20] Z Du et al. Approximations for a bottleneck steiner tree problem. *Algorithmica*, 32(4):554–561, 2002. 1, 2
- [21] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006. 1, 3
- [22] Joseph L Ganley and Jeffrey S Salowe. Optimal and approximate bottleneck steiner trees. *Operations Research Letters*, 19(5):217–224, 1996. 2
- [23] George Georgakopoulos and Christos H Papadimitriou. The 1-steiner tree problem. *Journal of Algorithms*, 8(1):122–130, 1987.
- [24] Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Parameterized complexity of generalized vertex cover problems. In *Proceedings of the 9th International Workshop Algorithms and Data Structures (WADS)*, volume 3608, pages 36–48. Springer, 2005. 3
- [25] Mathias Hauptmann and Marek Karpiński. *A compendium on Steiner tree problems*. Inst. für Informatik, 2013. 1

- [26] Yung-Tsung Hou, Chia-Mei Chen, and Bingchiang Jeng. An optimal new-node placement to enhance the coverage of wireless sensor networks. *Wireless Networks*, 16(4):1033–1043, 2010. [2](#)
- [27] Andrew B Kahng and Gabriel Robins. *On optimal interconnections for VLSI*, volume 301. Springer Science & Business Media, 1994. [1](#)
- [28] Richard M Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68, 1975. [1](#)
- [29] Klara Kedem, Ron Livne, János Pach, and Micha Sharir. On the union of jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete & Computational Geometry*, 1(1):59–71, 1986. [8](#), [9](#)
- [30] Zi-Mao Li, Da-Ming Zhu, and Shao-Han Ma. Approximation algorithm for bottleneck steiner tree problem in the euclidean plane. *Journal of Computer Science and Technology*, 19(6):791–794, 2004. [2](#)
- [31] Guo-Hui Lin and Guoliang Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, 69(2):53–57, 1999. [2](#)
- [32] Daniel Marx, Marcin Pilipczuk, and Michał Pilipczuk. On subexponential parameterized algorithms for Steiner Tree and Directed Subset TSP on planar graphs. *ArXiv e-prints*, July 2017. [arXiv:1707.02190](#). [3](#)
- [33] Nimrod Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM (JACM)*, 30(4):852–865, 1983. [13](#)
- [34] 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](#). [1](#)
- [35] Jesper Nederlof. Fast polynomial-space algorithms using inclusion-exclusion. *Algorithmica*, 65(4):868–884, 2013. [1](#), [3](#)
- [36] Marcin Pilipczuk, Michał Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Subexponential-time parameterized algorithm for Steiner tree on planar graphs. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 353–364, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. [3](#)
- [37] Marcin Pilipczuk, Michał Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Network sparsification for Steiner problems on planar and bounded-genus graphs. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 276–285. IEEE, 2014. [3](#)
- [38] Majid Sarrafzadeh and CK Wong. Bottleneck steiner trees in the plane. *IEEE Transactions on Computers*, 41(03):370–374, 1992. [2](#)
- [39] Lusheng Wang and Zimao Li. An approximation algorithm for a bottleneck k-steiner tree problem in the euclidean plane. *Information Processing Letters*, 81(3):151–156, 2002. [2](#)