The Complexity of Geodesic Spanners Using Steiner Points

Sarita de Berg
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Tim Ophelders
Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Irene Parada
Department of Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain

Frank Staals
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Jules Wulms
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

A geometric t-spanner $\mathcal{G}$ on a set S of n point sites in a metric space P is a subgraph of the complete graph on S such that for every pair of sites p, q the distance in $\mathcal{G}$ is a most t times the distance $d(p, q)$ in P. We call a connection between two sites a *link*. In some settings, such as when P is a simple polygon with m vertices and a link is a shortest path in P, links can consist of $\Theta(m)$ segments and thus have non-constant complexity. The spanner complexity is a measure of how compact a spanner is, which is equal to the sum of the complexities of all links in the spanner. In this paper, we study what happens if we are allowed to introduce k Steiner points to reduce the spanner complexity. We study such Steiner spanners in simple polygons, polygonal domains, and edge-weighted trees.

Surprisingly, we show that Steiner points have only limited utility. For a spanner that uses k Steiner points, we provide an $\Omega(nm/k)$ lower bound on the worst-case complexity of any $(3-\varepsilon)$-spanner, and an $\Omega(mn^{1/(t+1)}/k^{1/(t+1)})$ lower bound on the worst-case complexity of any $(t-\varepsilon)$-spanner, for any constant $\varepsilon \in (0, 1)$ and integer constant $t \geq 2$. These lower bounds hold in all settings. Additionally, we show NP-hardness for the problem of deciding whether a set of sites in a polygonal domain admits a 3-spanner with a given maximum complexity using k Steiner points.

On the positive side, for trees we show how to build a $2t$-spanner that uses k Steiner points of complexity $O(mn^{1/t}/k^{1/t} + n\log(n/k))$, for any integer $t \geq 1$. We generalize this result to forests, and apply it to obtain a $2\sqrt{2}t$-spanner in a simple polygon with total complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$. When a link in the spanner can be any path between two sites, we show how to improve the spanning ratio in a simple polygon to $(2k+\varepsilon)$, for any constant $\varepsilon \in (0, 2k)$, and how to build a $6t$-spanner in a polygonal domain with the same complexity.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases spanner, simple polygon, polygonal domain, geodesic distance, complexity

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2024.25

Related Version *Full Version*: https://arxiv.org/abs/2402.12110 [13]

Funding *Tim Ophelders*: partially supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.
Irene Parada: I. P. is a Serra Húnter Fellow. Partially supported by grant 2021UPC-MS-67392 funded by the Spanish Ministry of Universities and the European Union (NextGenerationEU) and by grant PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033.

Acknowledgements Work initiated at the 2023 AGA Workshop in Otterlo. We would like to thank Wouter Meulemans for his contributions in the initial phase of the project.

1 Introduction

Consider a set S of n point *sites* in a metric space P. In applications such as (wireless) network design [3], regression analysis [19], vehicle routing [12, 26], and constructing TSP tours [6], it is desirable to have a compact network that accurately captures the distances between the sites in S. Spanners provide such a representation. Formally, a *geometric t-spanner* $\mathcal{G}$ is a subgraph of the complete graph on S, so that for every pair of sites p, q the distance $d_{\mathcal{G}}(p, q)$ in $\mathcal{G}$ is at most t times the distance $d(p, q)$ in P [24]. The quality of a spanner can be expressed in terms of the *spanning ratio* t and a term to measure how "compact" it is. Typical examples are the *size* of the spanner, that is, the number of edges of $\mathcal{G}$, its weight (the sum of the edge lengths), or its diameter. Such spanners are well studied [4, 8, 10, 18]. For example, for point sites in $\mathbb{R}^d$ and any constant $\varepsilon > 0$ one can construct a $(1 + \varepsilon)$-spanner of size $O(n/\varepsilon^{d-1})$ [25]. Similar results exist for more general spaces [9, 20, 21]. Furthermore, there are various spanners with other desirable spanner properties such as low maximum degree, or fault-tolerance [7, 22, 23, 25].

When the sites represent physical locations, there are often other objects (e.g. buildings, lakes, roads, mountains) that influence the shortest path between the sites. In such settings, we need to explicitly incorporate the environment. We consider the case where this environment is modeled by a polygon P with m vertices, and possibly containing holes. The distance between two points $p, q \in P$ is then given by their *geodesic distance*: the length of a shortest path between p and q that is fully contained in P. This setting has been considered before. For example, Abam, Adeli, Homapou, and Asadollahpoor [1] present a $(\sqrt{10} + \varepsilon)$-spanner of size $O(n \log^2 n)$ when P is a simple polygon, and a $(5+\varepsilon)$-spanner of size $O(n\sqrt{h} \log^2 n)$ when the polygon has $h > 1$ holes. Abam, de Berg, and Seraji [2] even obtain a $(2 + \varepsilon)$-spanner of size $O(n \log n)$ when P is actually a terrain. To avoid confusion between the edges of P and the edges of $\mathcal{G}$, we will from hereon use the term *links* to refer to the edges of $\mathcal{G}$.

As argued by de Berg, van Kreveld, and Staals [15], each link in a geodesic spanner may correspond to a shortest path containing $\Omega(m)$ polygon vertices. Therefore, the *spanner complexity*, defined as the total number of line segments that make up all links in the spanner, more appropriate measures how compact a geodesic spanner is. In this definition, a line segment that appears in multiple links is counted multiple times: once for each link it appears in. The above spanners of [1, 2] all have worst-case complexity $\Omega(mn)$, hence de Berg, van Kreveld, and Staals present an algorithm to construct a $2\sqrt{2}t$-spanner in a simple polygon with complexity $O(mn^{1/t} + n \log^2 n)$, for any integer $t \geq 1$. By relaxing the restriction of links being shortest paths to any path between two sites, they obtain, for any constant $\varepsilon \in (0, 2t)$, a *relaxed* geodesic $(2t + \varepsilon)$-spanner in a simple polygon, or a relaxed geodesic $6t$-spanner in a polygon with holes, of the same complexity. These complexity bounds are still relatively high. De Berg, van Kreveld, and Staals [15] also show that these results are almost tight. In particular, for sites in a simple polygon, any geodesic $(3 - \varepsilon)$-spanner has worst-case complexity $\Omega(nm)$, and for any constant $\varepsilon \in (0, 1)$ and integer constant $t \geq 2$, a $(t - \varepsilon)$-spanner has worst-case complexity $\Omega(mn^{1/(t-1)} + n)$.

Problem Statement. A very natural question is then if we can reduce the total complexity of a geodesic spanner by allowing *Steiner points*. That is, by adding an additional set $\mathcal{S}$ of k vertices in $\mathcal{G}$, each one corresponding to a (Steiner) point in P. For the original sites $p, q \in S$ we still require that their distance in $\mathcal{G}$ is at most t times their distance in P, but the graph distance from a Steiner point $p' \in \mathcal{S}$ to any other site is unrestrained. Allowing for such Steiner points has proven to be useful in reducing the weight [5, 17] and size [22]

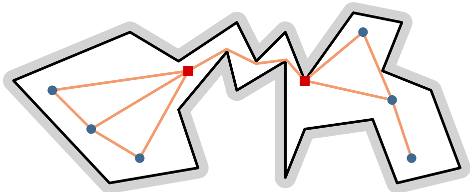

Figure 1 A spanner in a simple polygon that uses two Steiner points (red squares). By adding the two Steiner points, the spanner has a small spanning ratio and low complexity, as we no longer need multiple links that pass through the middle section of P.

of spanners. In our setting, it allows us to create additional "junction" vertices, thereby allowing us to share high-complexity subpaths. See Figure 1 for an illustration. Indeed, if we are allowed to turn every polygon vertex into a Steiner point, Clarkson [11] shows that, for any $\varepsilon > 0$, we can obtain a $(1+\varepsilon)$-spanner of complexity $O((n+m)/\varepsilon)$. However, the number of polygon vertices m may be much larger than the number of Steiner points we can afford. Hence, we focus on the scenario in which the number of Steiner points k is (much) smaller than m and n.

Our Contributions. Surprisingly, we show that in this setting, Steiner points have only limited utility. In some cases, even a single Steiner point allows us to improve the complexity by a linear factor. However, we show that such improvements are not possible in general. First of all, we show that computing a minimum cardinality set of Steiner points for sites in a polygonal domain that allow for a 3-spanner of a certain complexity is NP-hard. Moreover, we show that there is a set of n sites in a simple polygon with $m = \Omega(n)$ vertices for which any $(2-\varepsilon)$-spanner (with $k < n/2$ Steiner points) has complexity $\Omega(mn^2/k^2)$. Similarly, we give a $\Omega(mn/k)$ and $\Omega(mn^{1/(1+t)}/k^{1/(1+t)})$ lower bound on the complexity of a $(3-\varepsilon)$- and $(t-\varepsilon)$-spanner with k Steiner points. Hence, these results dash our hopes for a near linear complexity spanner with "few" Steiner points and constant spanning ratio.

These lower bounds actually hold in a more restricted setting. Namely, when the metric space is simply an edge-weighted tree that has m vertices, and the n sites are all placed in leaves of the tree. In this setting, we show that we can efficiently construct a spanner whose complexity is relatively close to optimal. In particular, our algorithm constructs a $2t$-spanner of complexity $O(mn^{1/t}/k^{1/t} + n\log(n/k))$. The main idea is to partition the tree into k subtrees of roughly equal size, construct a $2t$-spanner without Steiner points on each subtree, and connect the spanners of adjacent trees using Steiner points. The key challenge that we tackle, and one of the main novelties of the paper, is to make sure that each subtree contains only a constant number of Steiner points. We carefully argue that such a partition exists, and that we can efficiently construct it. Constructing the spanner takes $O(n\log(n/k) + m + K)$ time, where K is the output complexity. This output complexity is either the size of the spanner ($O(n\log(n/k))$), in case we only wish to report the endpoints of the links, or the complexity, in case we wish to explicitly report the shortest paths making up the links. An extension of this algorithm allows us to deal with a forest as well.

This algorithm for constructing a spanner on an edge-weighted tree turns out to be the crucial ingredient for constructing low-complexity spanners for point sites in polygons. In particular, given a set of sites in a simple polygon P, we use some of the techniques developed by de Berg, van Kreveld, and Staals [15] to build a set of trees whose leaves are the sites, and in which the distances in the trees are similar to the distances in the polygon. We then construct a $2t$-spanner with k Steiner points on this forest of trees using the above algorithm, and argue that this actually results into a $2\sqrt{2}t$-spanner with respect to the distances in the polygon. The main challenge here is to argue that the links used still have low complexity, even when they are now embedded in the polygon. We prove that the spanner (with respect to the polygon) has complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$, and can be constructed in time $O(n\log^2 n + m\log n + K)$. If we allow a link in the spanner to be any path between two sites (or Steiner points), then we obtain for any constant $\varepsilon \in (0, 2k)$ a relaxed $(2t+\varepsilon)$-spanner of the same complexity. For $k = O(1)$ our spanners thus match the results of de Berg, van Kreveld, and Staals [15]. Finally, we extend these results to polygonal domains, where we construct a similar complexity relaxed $6t$-spanner in $O(n\log^2 n + m\log n\log m + K)$ time.

Organization. We start with our results on edge-weighted trees in Section 2. To get a feel for the problem, we first establish lower bounds on the spanner complexity in Section 2.1. In Section 2.2 we present the algorithm for efficiently constructing a low complexity $2t$-spanner, in the full version [13] we extend it to a forest. In Section 3, we show how to use these results to obtain a $2\sqrt{2}t$-spanner for sites in a simple polygon P. In Section 4 we further extend our algorithms to the most general case in which P may even have holes. In the full version of the paper [13] we show that computing a minimum cardinality set of Steiner points with which we can simultaneously achieve a particular spanning ratio and maximum complexity is NP-hard. In Section 5 we pose some remaining open questions. Omitted proofs can be found in the full version [13].

2 Steiner spanners for trees

In this section, we consider spanners on an edge-weighted rooted tree T. We allow only positive weights. The goal is to construct a t-spanner on the leaves of the tree that uses k Steiner points, i.e. the set of sites S is the set of leaves. We denote by n the number of leaves and by m the number of vertices in T. The complexity of a link between two sites (or Steiner points) $p, q \in T$ is the number of edges in the shortest path $\pi(p, q)$, and the distance $d(p, q)$ is equal to the sum of the weights on this (unique) path. We denote by $T(v)$ the subtree of T rooted at vertex v. For an edge $e \in T$ with upper endpoint v_1 (endpoint closest to the root) and lower endpoint v_2, we denote by $T(e) := T(v_2) \cup \{e\}$ the subtree of T rooted at v_1.

The Steiner points are not restricted to the vertices of T, but can lie anywhere on the tree. To be precise, for any $\delta \in (0, 1)$ a Steiner point s can be placed on an edge (u, v) of weight w. This edge is then replaced by two edges (u, s) and (s, v) of weight δw and $(1-\delta)w$. Observe that this increases the complexity of a spanner on T by at most a constant factor as long as there are at most a constant number of Steiner points placed on a single edge. The next lemma states that it is indeed never useful to place more than one Steiner point on the interior of an edge.

▶ **Lemma 1.** *If a t-spanner $\mathcal{G}$ of a tree T has more than one Steiner point on the interior of an edge $e = (u, v)$, then we can modify $\mathcal{G}$ to obtain a t-spanner $\mathcal{G}'$ that has no Steiner points on the interior of e without increasing the complexity and number of Steiner points.*

Proof. Let $\mathcal{S}$ denote the set of Steiner points of $\mathcal{G}$ and let $\mathcal{S}(e) \subseteq \mathcal{S}$ the subset of Steiner points that lie on e. We assume that each Steiner point is used by a path $\pi_{\mathcal{G}}(p, q)$ for some sites p, q, otherwise we can simply remove it. We define the set of Steiner points of $\mathcal{G}'$ as $\mathcal{S}' = (\mathcal{S} \setminus \mathcal{S}(e)) \cup \{u, v\}$. Observe that $|\mathcal{S}'| \leq |\mathcal{S}|$. To obtain $\mathcal{G}'$, we replace each link (p, s) with $s \notin \mathcal{S}'$ by (p, u) if (p, s) intersects u and by (p, v) if (p, s) intersects v. Links between Steiner points on e are simply removed. Finally, we add the link (u, v) to $\mathcal{G}'$.

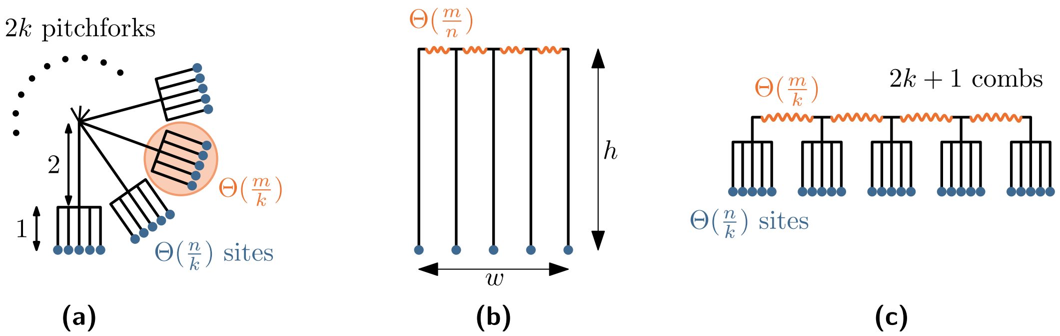

Figure 2 (a) Our construction for an $\Omega(mn^2/k^2)$ lower bound on the complexity of any $(2-\varepsilon)$-spanner. **(b)** A more detailed version of the comb of a pitchfork highlighted in the orange disk, which is also used for our $\Omega(mn^{1/(t+1)}/k^{1/(t+1)})$ lower bound on the complexity of any $(t-\varepsilon)$-spanner. **(c)** Our construction for an $\Omega(nm/k)$ lower bound on the complexity of any $(3-\varepsilon)$-spanner.

We first argue that the spanning ratio of $\mathcal{G}'$ is as most the spanning ratio of $\mathcal{G}$. Consider a path between two sites p, q in $\mathcal{G}$. If this path still exists in $\mathcal{G}'$, then $d_{\mathcal{G}}(p,q) = d_{\mathcal{G}'}(p,q)$. If not, then the path must visit e. Let (p_1, s_1) and (p_2, s_2) denote the first and last link in the path that connect to a Steiner point in the interior of e (possibly $s_1 = s_2$). If $\pi(p,q)$ does not intersect the open edge e, then these links are replaced by (p_1, u) and (p_2, u) (or symmetrically by (p_1, v) and (p_2, v)) in $\mathcal{G}'$. This gives a path in $\mathcal{G}'$ via u such that $d_{\mathcal{G}'}(p,q) < d_{\mathcal{G}}(p,q)$. If $\pi(p,q)$ does intersect e, i.e. p and q lie on different sides of e, then, without loss of generality, the links (p_1, s_1) and (p_2, s_2) are replaced by (p_1, u), (u, v), and (p_2, v). Again, this gives a path in $\mathcal{G}'$ such that $d_{\mathcal{G}'}(p,q) \leq d_{\mathcal{G}}(p,q)$.

Finally, what remains is to argue that the complexity of the spanner does not increase. Each link that we replace intersects either u or v, thus replacing this link by a link up to u or v reduces the complexity by one. Because each Steiner point on e occurs on at least one path between sites in $\mathcal{G}$, we replace at least two links. This decreases the total complexity by at least two, while including the edge (u, v) increases the complexity by only one. ◀

▶ **Corollary 2.** *Any spanner $\mathcal{G}$ on a tree T can be modified without increasing the spanning ratio and complexity such that no edge contains more than one Steiner point in its interior.*

2.1 Complexity lower bounds

In this section, we provide several lower bounds on the worst-case complexity of any $(t-\varepsilon)$-spanner that uses k Steiner points, where t is an integer constant and $\varepsilon \in (0,1)$. When Steiner points are not allowed, any $(2-\varepsilon)$-spanner in a simple polygon requires $\Omega(n^2)$ edges [1] and $\Omega(mn^2)$ complexity. If we allow a larger spanning ratio, say $(3-\varepsilon)$ or even $(t-\varepsilon)$, the worst-case complexity becomes $\Omega(mn)$ or $\Omega(mn^{1/(t-1)})$, respectively [15]. As the polygons used for these lower bounds are very tree-like, these bounds also hold in our tree setting. Next, we show how much each of these lower bounds is affected by the use of k Steiner points.

▶ **Lemma 3.** *For any constant $\varepsilon \in (0,1)$, there exists an edge-weighted tree T for which any $(2-\varepsilon)$-spanner using $k < n/2$ Steiner points has complexity $\Omega(mn^2/k^2)$.*

Proof sketch. The tree in Figure 2(a) is used to show this bound. Each of the k pitchforks that does not contain a Steiner point contributes $\Omega(mn^2/k^3)$ complexity to the spanner. ◀

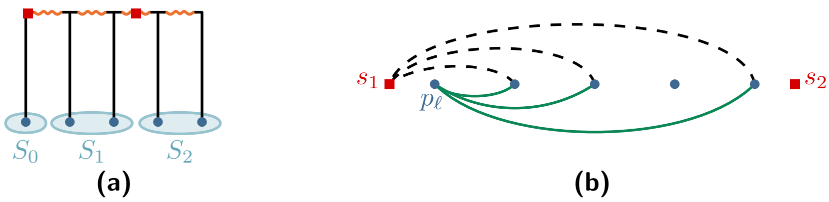

Figure 3 (a) The sets S_i defined by the two (red square) Steiner points. **(b)** For any spanner on S_1, every link to a Steiner point can be replaced by a link of smaller complexity, while increasing the spanning ratio by at most one. Here, the dashed links can be replaced by the green links.

▶ **Lemma 4.** *For any constant $\varepsilon \in (0,1)$, there exists an edge-weighted tree T for which any $(3-\varepsilon)$-spanner using $k < n/2$ Steiner points has complexity $\Omega(mn/k)$.*

Proof sketch. The tree in Figure 2(c) is used to show this lower bound. ◀

▶ **Lemma 5.** *For any constant $\varepsilon \in (0,1)$ and integer constant $t \geq 2$, there exists an edge-weighted tree T for which any $(t-\varepsilon)$-spanner using $k < n$ Steiner points has complexity $\Omega(mn^{1/(t+1)}/k^{1/(t+1)})$.*

Before we prove Lemma 5, we first discuss a related result in a simpler metric space. Let ϑ_n be the 1-dimensional Euclidean metric space with n points $v_1, \dots, v_n$ on the x-axis at $1, 2, \dots, n$. A link (v_i, v_j) has complexity $|i-j|$. Dinitz, Elkin, and Solomon [16] give a lower bound on the total complexity of any spanning subgraph of ϑ_n, given that the link-radius is at most ρ. The link-radius (called hop-radius in [16]) $\rho(G, r)$ of a graph G with respect to a root r is defined as the maximum number of links needed to reach any vertex in G from r. The link-radius of G is then $\min_{r \in V} \rho(G, r)$. The link-radius is bounded by the link-diameter, which is the minimum number of links that allow reachability between any two vertices.

▶ **Lemma 6** (Dinitz et al. [16]). *For any sufficiently large integer n and positive integer $\rho < \log n$, any spanning subgraph of ϑ_n with link-radius at most ρ has complexity $\Omega(\rho \cdot n^{1+1/\rho})$.*

Proof of Lemma 5. Consider the tree construction illustrated in Figure 2(b). This edge-weighted tree T has the shape of a comb of width w and height h with n teeth separated by corridors of complexity $M = \Theta(m/n)$ each. Each leaf at the bottom of a comb tooth is a site.

Any spanning subgraph of ϑ_n of complexity C and link-diameter ρ is in one-to-one correspondence with a $(\rho+1-\varepsilon)$-spanner of complexity $C \cdot m/n$ in T [15]. Lemma 6 then implies that any $(t-\varepsilon)$-spanner has worst-case complexity $\Omega(mn^{1/(t-1)})$.

When a set $\mathcal{S}$ of k Steiner points is introduced, we consider the at most $k+1$ sets $S_0, \dots, S_k$ of consecutive sites that have no Steiner point between them; see Figure 3(a). We can replace any link (p, q) where $p, q \in S \cup \mathcal{S}$ and $\pi(p, q)$ intersects a Steiner point s by the links (p, s) and (s, q). Corollary 2 implies that this increases the complexity by only a constant factor. From now on, we thus assume there are no such links. We claim the following for any $(t-\varepsilon)$-spanner $\mathcal{G}$ on $S = S_0 \cup \dots \cup S_k$.

▷ Claim 7. Let C_i be the complexity of the subgraph of $\mathcal{G}$ induced by S_i and the at most two Steiner points s_ℓ and s_r bounding S_i from the left and right, respectively. Then, we can construct a $(t+2-\varepsilon)$-spanner $\mathcal{G}'_i$ on S_i that has complexity at most C_i.

Proof of Claim. Let p_ℓ and p_r denote the leftmost and rightmost site in S_i. We replace each link (p, s_ℓ) (or (p, s_r)), $p \in S_i$, by the link (p, p_ℓ) (resp. (p, p_r)). If there is a link (s_ℓ, s_r), it is replaced by (p_ℓ, p_r). Any path in $\mathcal{G}$ between $p, q \in S_i$ that visits either s_ℓ and/or s_r

corresponds to a path via p_r and/or p_ℓ in $\mathcal{G}'_i$. The length of the path increases by $2h$ when visiting p_r or p_ℓ, so by at most $4h$ when visiting both. As $d(p,q) \geq 2h$, the spanning ratio increases by at most two. ◁

These changes in the spanner only decrease the complexity of the subspanner on S_i. Notice also that if we apply them to each of the sets S_i, each link of $\mathcal{G}$ is changed by only one of the subspanners $\mathcal{G}'_i$. Thus, we consider the minimum complexity of any $(t+2-\varepsilon)$-spanner on these sites. By applying Lemma 6, we find that the worst-case complexity of any $(t+2-\varepsilon)$-spanner on these $|S_i|$ sites is $\Omega(m/n \cdot |S_i|^{1+1/(t+1)})$. The complexity of $\mathcal{G}$ is at least the sum of the complexities of these $\mathcal{G}'_i$ spanners over all S_i, so $\frac{m}{n}\sum_{i=0}^{k}\Omega\left(|S_i|^{1+1/(t+1)}\right)$, where $\sum_{i=0}^{k}|S_i| = \Theta(n)$. Using a logarithmic transformation and induction, we see that this sum is minimized when $|S_i| = \Theta(n/k)$ for all $i \in 0,\dots,k$. So,

$$\frac{m}{n}\sum_{i=0}^{k}\Omega\left(|S_i|^{1+1/(t+1)}\right) \geq \frac{m}{n}\sum_{i=0}^{k}\Omega\left((n/k)^{1+1/(t+1)}\right) = \Omega\left(mn^{1/(t+1)}/k^{1/(t+1)}\right). \blacktriangleleft$$

2.2 A low complexity Steiner spanner

In this section, we describe how to construct low complexity spanners for edge-weighted trees. The goal is to construct a $2t$-spanner of complexity $O(mn^{1/t}/k^{1/t} + n\log n)$ that uses at most k Steiner points. We first show that the spanner construction for a simple polygon of [15] can be used to obtain a low complexity spanner for a tree (without Steiner points).

▶ **Lemma 8** (de Berg et al. [15]). *For any integer $t \geq 1$, we can build a $2t$-spanner for an edge-weighted tree T of size $O(n\log n)$ and complexity $O(mn^{1/t} + n\log n)$ in $O(n\log n + m)$ time.*

Spanner construction. Given an edge-weighted tree T, to construct a Steiner spanner $\mathcal{G}$ for T, we start by partitioning the sites in k sets $S_1,\dots S_k$ by an in-order traversal of the tree. The first $\lceil n/k\rceil$ sites encountered are in S_1, the second $\lceil n/k\rceil$ in S_2, etc. After this, the sites are reassigned into k new disjoint sets $S'_1,\dots S'_k$. For each of these sets, we consider a subtree $T'_i \subseteq T$ whose leaves are the set S'_i. There are four properties that we desire of these sets and their subtrees.

1. The size of S'_i is $O(n/k)$.
2. The trees T'_i cover T, i.e. $\bigcup_i T'_i = T$.
3. The trees T'_i are disjoint apart from Steiner points.
4. Each tree T'_i contains only $O(1)$ Steiner points.

As we prove later, these properties ensure that we can construct a spanner on each subtree T'_i to obtain a spanner for T. We obtain such sets S'_i and the corresponding trees T'_i as follows.

We color the vertices and edges of the tree T using k colors $\{1,\dots,k\}$ in two steps. In this coloring, an edge or vertex is allowed to have more than one color. First, for each set S_i, we color the smallest subtree that contains all sites in S_i with color i. After this step, all uncolored vertices have only uncolored incident descendant edges. Second, we color the remaining uncolored edges and vertices. These edges and their (possibly already colored) upper endpoints are colored in a bottom-up fashion. We assign each uncolored edge and its upper endpoint the color with the lowest index i that is assigned also to its lower endpoint.

After coloring T, we for $i \in \{1,\dots,k\}$ place a Steiner point s_i at the root of tree T_i formed by all edges and vertices of color i. This may place multiple Steiner points at the same vertex. We may abuse notation, and denote by s_i the vertex occupied by Steiner point s_i.

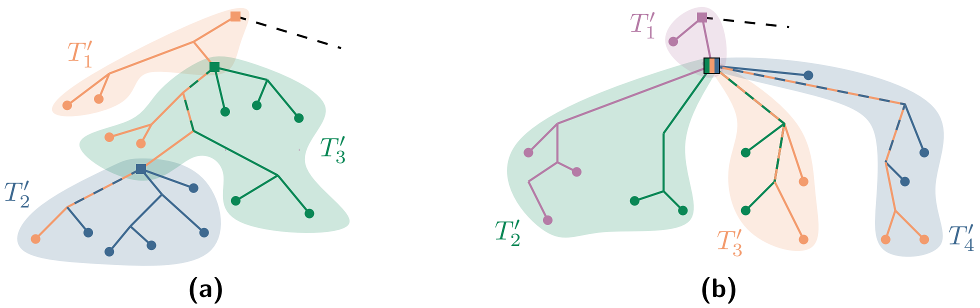

Figure 4 The tree T_i is the subtree whose edges and vertices have color i. A Steiner point (square) is placed at the root of T_i. The shaded areas show the trees T_i'. The examples show the case when the Steiner points are **(a)** at different vertices or **(b)** share a vertex.

For each Steiner point s_i, we define a subtree $T_i' \subseteq T$. The sites in T_i' will be the set S_i'. The tree T_i' is a subtree of $T(s_i)$. When s_i is the only Steiner point at the vertex, then $T_i' = T(s_i) \setminus \bigcup_j (T(s_j) \setminus \{s_j\})$ for s_j a descendant of s_i. In other words, we look at the tree rooted at s_i up to and including the next Steiner points, see Figure 4(a). When s_i is not the only Steiner point at the vertex, we include only subtrees $T(e)$ of s_i (up to the next Steiner points) that start with an edge e that has color i and no color $j > i$. See Figure 4(b). Whenever s_i has the lowest or highest index of the Steiner points at s_i, we also include all $T(e')$ that start with an edge e' of color $j < i$ or $j > i$, respectively. This generalizes the scheme for when s_i is the only Steiner point at the vertex.

By creating T_i' in this way, s_i is not a leaf of T_i'. We therefore adapt T_i' by adding an edge of weight zero between the vertex at s_i and a new leaf corresponding to s_i. On each subtree T_i', we construct a $2t$-spanner using the algorithm of Lemma 8. These k spanners connect at the Steiner points, which we formally prove in the spanner analysis.

Analysis. To prove that $\mathcal{G}$ is indeed a low complexity $2t$-spanner for T, we first show that the four properties stated before hold for S_i' and T_i'. We often apply the following lemma, that limits the number of colors an edge can be assigned by our coloring scheme.

▶ **Lemma 9.** *An edge can have at most two colors.*

Proof. First, observe that an edge can receive more than one color only in the first step of the coloring. Suppose for contradiction that there is an edge e in T that has three colors $i < j < \ell$. Let v be the lower endpoint of e. Then there must be three sites $p_i \in S_i$, $p_j \in S_j$, $p_\ell \in S_\ell$ in $T(v)$. Because these sets are defined by an in-order traversal, p_i must appear before p_j in the traversal. Similarly, p_j appears before p_ℓ. Additionally, there must be a site $p_j' \in S_j$ in $T \setminus T(v)$, otherwise the color j would not be assigned to e. The site p_j' must appear before p_i or after p_ℓ in the traversal. In the first case, p_i must be in S_j as it appears between two sites in S_j. In the second case, we find $p_\ell \in S_j$, also giving a contradiction. ◀

We prove properties 1, 2, and 3 in the full version [13].

▶ **Lemma 10.** *There are at most five Steiner points in T_i'.*

Proof sketch. By definition, s_i is in T_i', so we want to show that there are at most four other Steiner points in T_i'. Note that a Steiner point can occur in T_i' only if its path to s_i does not encounter any other Steiner point. In this proof sketch we show there are at most

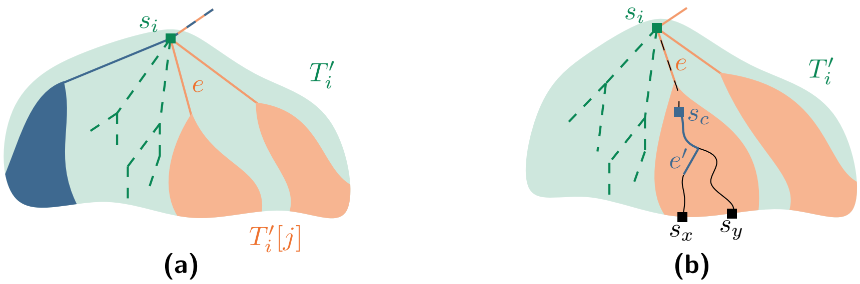

Figure 5 Notation used in Lemma 10. In $T_i'[j]$ are all subtrees that start with an edge of color j.

two Steiner points in subtrees $T(e)$ for which the edge e, which is incident to s_i in T_i', does not have color i. In the full proof we use similar techniques to bound the number of Steiner points in subtrees for which e does have color i by two as well.

Let $T_i'[j]$ be the subtree of T_i' rooted at s_i that is the union of $T(e) \cap T_i'$ for all edges e incident to s_i of color $j \neq i$ and not of color i as well, see Figure 5(a). We argue that this subtree is non-empty for at most two colors j. Consider such an edge e. Because e does not have color i and $e \in T_i'$, it must be that s_j is above s_i in T. Thus, the parent edge of s_i in T must also be colored j. By Lemma 9, the parent edge of s_i in T can be assigned at most two colors, so $T_i'[j]$ is non-empty for at most two colors.

Next, we prove that $T_i'[j]$ contains at most one Steiner point other than s_i. We assume that $i < j$, the proof for $i > j$ is symmetric. Assume for contradiction that $T_i'[j]$ contains two Steiner points s_x and s_y, $x < y$; see Figure 5(b). As shown before, there is a site of S_j in $T \setminus T(s_i)$. As $i < j$, this implies that $i < x < y < j$. Let e' be the first edge on $\pi(s_i, s_x)$ that is not on $\pi(s_i, s_y)$, i.e. the first edge after the paths diverge. Let c be a color of e' and let v and w be the upper and lower endpoint of e'. The tree $T(w)$ does not contain any sites of S_j, as these appear in the traversal after the sites of S_x. It follows that S_c is before S_j in the in-order traversal, in other words $i < c < j$. The parent edge of s_i cannot be colored c, as a site of S_c would then appear either before a site in S_i or after a site in S_j in the in-order traversal. It follows that s_c is on the path $\pi(v, s_i)$. If $s_c \neq s_i$, this contradicts the assumption that this path does not contain a Steiner point. If $s_c = s_i$, then $i < c$ implies that the subtree starting with an edge of color j is in not T_i', which is a contradiction. We conclude that there are at most two Steiner points in the subtrees $T_i'[j]$ in total for all $j \neq i$. ◀

We are now ready to prove that our algorithm computes a spanner with low complexity.

▶ **Lemma 11.** *The spanner $\mathcal{G}$ is a $2t$-spanner for T of size $O(n \log(n/k))$ and complexity $O(mn^{1/t}/k^{1/t} + n \log(n/k))$.*

Proof. To bound the size and complexity of the spanner, we first consider the number of leaves n_i and vertices m_i in each subtree T_i'. As n_i is equal to $|S_i'|$ plus the number of Steiner points in T_i', properties 1 and 4 (Lemma 10) imply that $n_i = O(n/k) + 5 = O(n/k)$. Property 3 states the subtrees T_i' are disjoint apart from their shared Steiner points, so $\sum m_i = O(m)$. By Lemma 8, $\mathcal{G}$ has size $\sum_{i=1}^{k} O\left(\frac{n}{k} \log\left(\frac{n}{k}\right)\right) = O\left(n \log\left(\frac{n}{k}\right)\right)$ and complexity $\sum_{i=1}^{k} O\left(m_i \left(\frac{n}{k}\right)^{1/t} + \frac{n}{k} \log\left(\frac{n}{k}\right)\right) = O\left(\frac{mn^{1/t}}{k^{1/t}} + n \log\left(\frac{n}{k}\right)\right)$.

What remains is to show that $\mathcal{G}$ is a $2t$-spanner. Let $p, q \in S$ be two leaves in T. If $p, q \in S_i'$ for some $i \in \{1, \dots, k\}$ then the shortest path $\pi(p, q)$ is contained within T_i'. The $2t$-spanner on T_i' implies that $d_{\mathcal{G}}(p, q) \leq 2td(p, q)$. If there is no such set S_i' that contains both sites, consider the sequence of vertices $v_1, \dots, v_\ell$ where $\pi(p, q)$ exits some subtree T_i'. Let v, w be two consecutive vertices in this sequence. Without loss of generality, assume that $w \in T(v)$, and let s_x be the Steiner point at v for which $w \in T'_x$ (properties 2 and 3 imply s_x exists). Then the $2t$-spanner on T'_x ensures that $d_{\mathcal{G}}(v, w) \leq 2td(v, w)$. It follows that $d_{\mathcal{G}}(p, q) \leq d_{\mathcal{G}}(p, v_1) + d_{\mathcal{G}}(v_1, v_2) + \cdots + d_{\mathcal{G}}(v_\ell, q) \leq 2t(d(p, v_1) + d(v_1, v_2) + \cdots + d(v_\ell, q)) = 2td(p, q)$. ◀

▶ **Theorem 12.** *Let T be a tree with n leaves and m vertices, and $t \leq 1$ be any integer constant. For any $1 \leq k \leq n$, we can build a $2t$-spanner $\mathcal{G}$ for T using at most k Steiner points of size $O(n \log(n/k))$ and complexity $O(mn^{1/t}/k^{1/t} + n \log(n/k))$ in $O(n \log(n/k) + m + K)$ time, where K is the output size.*

A forest spanner. The tree spanner can be extended to a spanner for a forest $\mathcal{F}$. As $\mathcal{F}$ is disconnected, we cannot require all sites to have a path between them in the spanner. Instead, we say that $\mathcal{G}$ is a t-spanner for $\mathcal{F}$ if $\mathcal{G}$ is a t-spanner for every tree in $\mathcal{F}$.

▶ **Theorem 13.** *Let $\mathcal{F}$ be a forest with n leaves and m vertices, and $t \leq 1$ be any integer constant. For any $1 \leq k \leq n$, we can build a $2t$-spanner $\mathcal{G}$ for $\mathcal{F}$ using at most k Steiner points of size $O(n \log(n/k))$ and complexity $O(mn^{1/t}/k^{1/t} + n \log(n/k))$ in $O(n \log(n/k) + m + K)$ time, where K is the output size.*

3 Steiner spanners in simple polygons

We consider the problem of computing a t-spanner using k Steiner points for a set of n point sites in a simple polygon P with m vertices. We measure the distance between two points in p, q in P by their geodesic distance, i.e. the length of the shortest path $\pi(p, q)$ fully contained within P. A link (p, q) in the spanner is the shortest path $\pi(p, q)$, and its complexity is the number of segments in this path. Lower bounds for trees straightforwardly extend to polygonal instances. Again, we aim to obtain a spanner of complexity close to the lower bound.

▶ **Lemma 14.** *The lower bounds of Lemmata 3, 4, and 5 also hold for simple polygons.*

Spanner construction. Next, we describe how to obtain a low-complexity spanner in a simple polygon using at most k Steiner points. In our approach, we combine ideas from [2] and [15] with the forest spanner of Theorem 13. We first give a short overview of the approach to obtain a low complexity $2\sqrt{2}t$-spanner [15], and then discuss how to combine these ideas with the forest spanner to obtain a low complexity Steiner spanner.

We partition the polygon P into two subpolygons P_ℓ and P_r by a vertical line segment λ such that roughly half of the sites lie in either subpolygon. For the line segment λ, we then consider the following weighted 1-dimensional space. For each site $p \in S$, let p_λ be the *projection* of p: the closest point on λ to p. The (weighted 1-dimensional distance) between two sites p_λ, q_λ is defined as $d_w(p_\lambda, q_\lambda) := d(p, p_\lambda) + d(p_\lambda, q_\lambda) + d(q, q_\lambda)$. In other words, the sites in the 1-dimensional space are weighted by the distance to their original site in P. For this 1-dimensional space we construct a t-spanner $\mathcal{G}_\lambda$, and for each link (p_λ, q_λ) in $\mathcal{G}_\lambda$ we add the link (p, q) to the spanner $\mathcal{G}$. Finally, we process the subpolygons P_ℓ and P_r recursively. De Berg, van Kreveld, and Staals [15] show that this gives a $\sqrt{2}t$-spanner in a simple polygon. To obtain a spanner of complexity $O(mn^{1/t} + n \log^2 n)$, they construct a 1-dimensional $2t$-spanner $\mathcal{G}_\lambda$ using the approach of Lemma 8, resulting in a $2\sqrt{2}t$-spanner.

In our case, we require information on the paths from the sites to their projection instead of only their distance to decide where to place the Steiner points. This information is captured in the shortest path tree SPT_λ of the segment λ, which is the union of all shortest

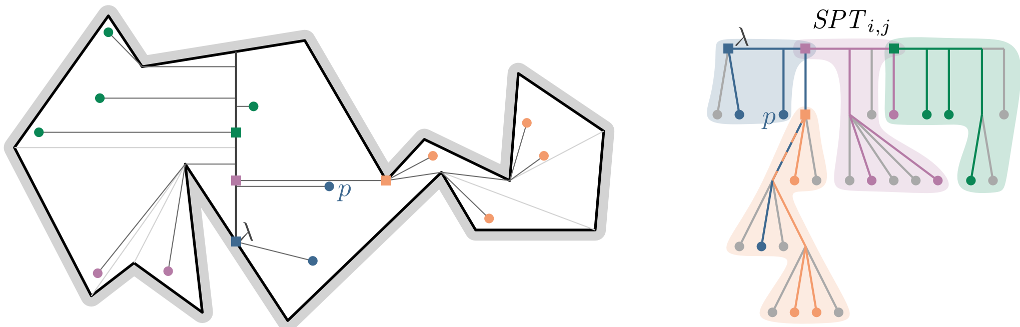

Figure 6 The shortest path tree of λ in P' and its $SPT_{i,j}$. The grey nodes and edges are not included in $SPT_{i,j}$, but can be assigned to a T'_i as indicated by the colored backgrounds. The squares show the Steiner points in $SPT_{i,j}$ and P'. The sites in P' are colored as the trees T'_i.

paths from the vertices of P to their closest point on λ. Additionally, we include all sites in S in the tree SPT_λ. The segment λ is split into multiple edges at the projections of the sites, see Figure 6. The tree SPT_λ is rooted at the lower endpoint of λ and has $O(m+n)$ vertices.

We adapt the algorithm to build a spanner in P as follows. Instead of computing a 1-dimensional spanner directly in each subproblem in the recursion, we first collect the shortest path trees of all subproblems. Let $SPT_{i,j}$ denote the shortest path tree of the j-th subproblem at the i-th level of the recursion. We exclude all vertices from $SPT_{i,j}$ that have no site as a descendant. This ensures that all leaves of the tree are sites. Let $\mathcal{F} = \cup_{i,j} SPT_{i,j}$ be the forest consisting of all trees. A site in S or vertex of P can occur in multiple trees $SPT_{i,j}$, but they are seen as distinct sites and vertices in the forest $\mathcal{F}$. We call a tree $SPT_{i,j}$ *large* if $0 \le i \le \log k$ and *small* otherwise. In other words, the trees created in the recursion up to level $\log k$ are large. We then partition $\mathcal{F}$ into two forests $\mathcal{F}_s$ and $\mathcal{F}_\ell$ containing the small and large trees. For each tree in $\mathcal{F}_s$ we directly apply the $2t$-spanner of Lemma 8 that uses no Steiner points to obtain a spanner $\mathcal{G}_s$. For the forest $\mathcal{F}_\ell$ we apply Theorem 13 to obtain a $2t$-spanner $\mathcal{G}_\ell$ for $\mathcal{F}_\ell$. Let $\mathcal{G}_\mathcal{F} = \mathcal{G}_s \cup \mathcal{G}_\ell$. A Steiner point in $\mathcal{G}_\mathcal{F}$ corresponds to either a vertex of P or a point on λ. Let $\mathcal{S}$ denote the set of Steiner points. To obtain a spanner $\mathcal{G}$ in the simple polygon, we add a link (p,q), $p,q \in S \cup \mathcal{S}$, to $\mathcal{G}$ whenever there is a link in $\mathcal{G}_\mathcal{F}$ between (a copy of) p and q.

▶ **Lemma 15.** *The graph $\mathcal{G}$ is a $2\sqrt{2}t$-spanner for the sites S in P of size $O(n \log^2 n)$.*

Complexity analysis. To bound the complexity of the links in $\mathcal{G}$, we have to account for the complexity of links generated by both $\mathcal{G}_s$ and $\mathcal{G}_\ell$. Bounding the complexity of $\mathcal{G}_s$ is relatively straightforward, but to bound the complexity of $\mathcal{G}_\ell$ we first prove a lemma on the structure of a shortest path in P between sites in $\mathcal{F}_\ell$.

Let T be a tree in $\mathcal{F}_\ell$ and let P' be the corresponding subpolygon of the subproblem. We consider the shortest path between two sites that are assigned to the same subtree T'_i of T by the forest algorithm. It can be that this shortest path uses vertices of P' that were excluded from T, as they had no site as a descendant. For the analysis, we do include these vertices in T and assign them to subtrees T'_j as in Section 2.2; see Figure 6. The following lemma states that the complexity of a shortest path between two sites in the same subtree T'_i is bounded by the number of vertices in T'_i. We use this to bound the complexity in Lemma 17.

▶ **Lemma 16.** *A shortest path $\pi(p,q)$ in P' between sites $p,q \in T'_i$ uses vertices in T'_i only.*

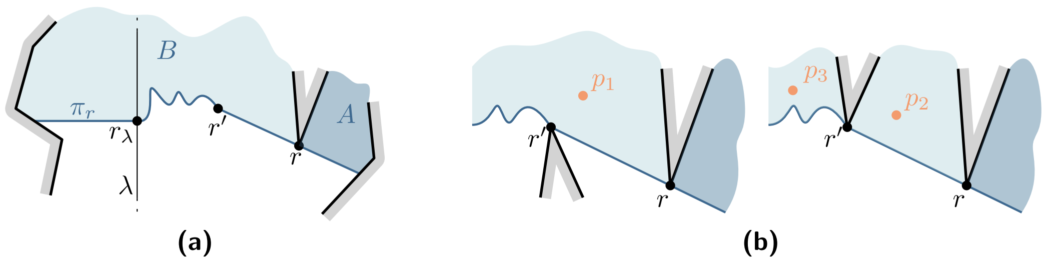

Figure 7 (a) The extended path π_r separates the polygon into $P'_r = A \cup B$ and $P'_{\neg r}$. **(b)** Sites $p_{1\ldots 3}$ correspond to the respective subcases (i–iii) based on the structure of the polygon around r'.

Proof. Assume for contradiction that r is a highest vertex in T used by $\pi(p,q)$ that is not in T'_i. First, consider the case that r is the root of T. Recall that this means r is the bottom endpoint of λ, and thus lies on the boundary of P'. As r is on $\pi(p,q)$, it must be that p and q lie in different subpolygons, and at least one of them lies below the horizontal line through r. This implies that $s_i = r$, which is a contradiction.

Next, consider the case that r is not the root of T. Let r' be the parent of r. If r' is in T'_i, then it must be a leaf. We consider the following partition of P'. Recall that r_λ denotes the closest point on λ to r. We extend the shortest path $\pi(r, r_\lambda)$ to the boundary of P' by extending the first and last line segments of the path to obtain a path π_r, see Figure 7(a). Let $\partial P'$ denote the boundary of P'. We define P'_r to be the closed polygon bounded by $\partial P'$ and π_r that contains the polygon edges incident to r, and $P'_{\neg r} := P' \setminus P'_r$. Because r is a reflex vertex of P', P'_r is well-defined. Without loss of generality, we assume that P'_r contains the part of λ above r_λ, as in Figure 7(a). If both p and q are in $P'_{\neg r}$, then $r \notin \pi(p,q)$. It follows that p and/or q are in P'_r. Without loss of generality, assume that $p \in P'_r$.

We distinguish two cases based on the location of p, see Figure 7(a). Either $p \in A$, where $A \subset P'_r$ is bounded by the extension segment starting at r and $\partial P'$, or $p \in B$, where $B \subset P'_r$ is bounded by $\pi(r, r_\lambda)$, the extension segment starting at r_λ, and $\partial P'$.

If $p \in A$, then p is a descendant of r in T. As p and q are in T'_i and r is not, it must be that q is also a descendant of r. It follows that $q \in A$, but this means that r is not a reflex vertex on $\pi(p,q)$, which contradicts it being a shortest path.

If $p \in B$, the previous paragraph implies that $q \notin A$. Additionally, $q \notin B$ as well, as r would then not be a reflex vertex in $\pi(p,q)$. It follows that $q \in P'_{\neg r}$. Next, we make a distinction on whether r' is a vertex of P' or not. First, assume that r' is not a vertex of P', and thus $r' \in \lambda$. Because $p \in B$, p_λ must be at or above r'. Because $q \in P'_{\neg r}$, q_λ must be below r'. This implies that the path in T from p to q visits r', which contradicts $p, q \in T'_i$.

Next, we assume that r' is a vertex of P'. We distinguish three different subcases based on the shape of the polygon around r', see Figure 7(b), and find a contradiction in each case:

(i) The edges of P' incident to r' are in $P'_{\neg r}$. As r is on $\pi(p,q)$, q must be a descendant of r'. It follows that the Steiner point s_i is located on the path in T from q to r', so p is also a descendant of r'. It follows that p is on the segment rr'. However, for r to be on $\pi(p,q)$, q must then be in A, which is a contradiction.

(ii) The edges of P' incident to r' are in P'_r, and r' is on $\pi(p, p_\lambda)$. In this case, p is a descendant of r'. This again implies that q is a descendent of r', which contradicts $q \in P'_{\neg r}$.

(iii) The edges of P' incident to r' are in P'_r, and r' is not on $\pi(p, p_\lambda)$. The path $\pi(p,q)$ either intersects the boundary of B twice, which is not allowed as both are shortest paths, or visits r' as well. However, this implies that $q \in A$, which is a contradiction. ◀

▶ **Lemma 17.** *The spanner $\mathcal{G}$ has complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n \log^2 n)$.*

Proof. To bound the complexity of the links in $\mathcal{G}$ generated by $\mathcal{G}_s$ we apply Lemma 8 directly. As Lemma 8 corresponds to the algorithm to construct a low complexity spanner in a polygon using the shortest path tree, the complexity bound also holds in the simple polygon setting. Using $\sum_{j=0}^{2^i} m_{i,j} = O(m)$, where $m_{i,j}$ is the number of vertices in $SPT_{i,j}$, the complexity is

$$\sum_{i=\log k}^{O(\log n)} \sum_{j=0}^{2^i} O\left(m_{i,j}\left(\frac{n}{2^i}\right)^{1/t} + \frac{n}{2^i}\log\left(\frac{n}{2^i}\right)\right) = O\left(\frac{mn^{1/t}}{k^{1/t}} + n\log^2 n\right).$$

For $\mathcal{F}_\ell$, the algorithm of Lemma 8 is used as a subroutine on every subtree T_i'. Lemma 16 implies that the complexity bound of Theorem 13 also holds for links in P. Recall that the number of sites in $\mathcal{F}_\ell$ is $O(n\log k)$. A vertex of P can occur in at most two subproblems at each level of the recursion that partitions P, thus the number of vertices in $\mathcal{F}_\ell$ is $O((m+n)\log k)$. As the n sites are equally divided over all subproblems at level i, the complexity of the links in $\mathcal{G}$ generated by $\mathcal{G}_\ell$ given by Theorem 13 is improved to

$$O\left(\frac{m\log k(n\log k)^{1/t}}{k^{1/t}} + n\log k\log\left(\frac{n\log k}{k}\right)\right) = O\left(\frac{mn^{1/t}(\log k)^{1+1/t}}{k^{1/t}} + n\log^2 n\right). \blacktriangleleft$$

▶ **Theorem 18.** *Let S be a set of n point sites in a simple polygon P with m vertices, and $t \geq 1$ be any integer constant. For any $1 \leq k \leq n$, we can build a geodesic $2\sqrt{2}t$-spanner with at most k Steiner points, of size $O(n\log^2 n)$ and complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$ in $O(n\log^2 n + m\log n + K)$ time, where K is the output size.*

A relaxed geodesic $(2k+\varepsilon)$-spanner. In a more recent version of the paper by de Berg, van Kreveld, and Staals [14, 15] they show how to apply the refinement proposed by Abam, de Berg, and Seraji [2] to improve the spanning ratio to $(2k+\varepsilon)$ for any constant $\varepsilon \in (0, 2k)$. They make two changes in their approach. First, instead of using the shortest path between two sites as a link they allow a link to be any path between two sites. They call such a spanner a *relaxed* geodesic spanner. Second, for each split of the polygon they construct spanners on several sets of sites in the 1-dimensional weighted space. Using the same adaptations, we obtain a relaxed $(2k+\varepsilon)$-spanner of complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$.

4 Steiner spanners in polygonal domains

If the polygon contains holes, the spanner construction in the previous section no longer suffices. In particular, we may need a different type of separator, and shortest paths in P are no longer restricted to vertices in some subtree (Lemma 16 does not hold). De Berg, van Kreveld, and Staals [15] run into similar problems when generalizing their low complexity spanner, and solve them as follows. There are two main changes in their construction. First, the separator is no longer a line segment, but a *balanced separator* that consists of at most three shortest paths that partition the domain into two subdomains P_r and P_ℓ. They then construct a spanner $\mathcal{G}_\lambda$ on the 1-dimensional space containing the projections of the sites for each shortest path in the separator. Second, the links that are included in the spanner are no longer shortest paths, but consist of at most three shortest paths, resulting in a relaxed geodesic spanner. In contrast to the simple polygon, using a 1-dimensional spanner with spanning ratio t results in a spanning ratio in P of $3t$ [15].

To construct a low complexity spanner using k Steiner points, we use our simple polygon approach with the adaptions of [15]. The number of trees, and thus the number of sites and vertices in the trees, increases by a constant factor, as we create at most three shortest path trees at each level. To bound the complexity, we can no longer apply Lemma 16. However, the links that are added to $\mathcal{G}$ are shortest paths in the shortest path tree. Therefore, the bound on the complexity of $\mathcal{G}_{\mathcal{F}}$ directly translates to a bound on the complexity of $\mathcal{G}$. As in the simple polygon case, we obtain a spanner of complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$.

▶ **Theorem 19.** *Let S be a set of n point sites in a polygonal domain P with m vertices, and $t \geq 1$ be any integer constant. For any $k \leq n$, we can build a relaxed geodesic $6t$-spanner with at most k Steiner points, of size $O(n\log n\log(n/k))$ and complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$ in $O(n\log^2 n + m\log n\log m + K)$ time, where K is the output size.*

5 Future work

On the side of constructing low-complexity spanners, an interesting direction for future work would be to close the gap between the upper and lower bounds, both with and without using Steiner points. We believe it might be possible to increase the $n^{1/(t+1)}$ term to $n^{1/t}$ (or even $n^{1/(t-1)}$) in Lemma 5. On the side of the hardness, many interesting open questions remain, such as: Is the problem still hard in a simple polygon? Can we show hardness for other spanning ratios and/or a less restricted complexity requirement? Is the problem even in NP?

References

1 Mohammad Ali Abam, Marjan Adeli, Hamid Homapour, and Pooya Zafar Asadollahpoor. Geometric spanners for points inside a polygonal domain. In *Proc. 31st International Symposium on Computational Geometry, SoCG*, volume 34 of *LIPIcs*, pages 186–197. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.SOCG.2015.186.

2 Mohammad Ali Abam, Mark de Berg, and Mohammad Javad Rezaei Seraji. Geodesic spanners for points on a polyhedral terrain. *SIAM J. Comput.*, 48(6):1796–1810, 2019. doi:10.1137/18M119358X.

3 Khaled M. Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Trans. Parallel Distributed Syst.*, 14(4):408–421, 2003. doi:10.1109/TPDS.2003.1195412.

4 Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. 27th Annual ACM Symposium on Theory of Computing, STOC*, pages 489–498. ACM, 1995. doi:10.1145/225058.225191.

5 Sujoy Bhore and Csaba D. Tóth. Light Euclidean Steiner spanners in the plane. In *Proc. 37th International Symposium on Computational Geometry, SoCG*, volume 189 of *LIPIcs*, pages 15:1–15:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.SOCG.2021.15.

6 Glencora Borradaile and David Eppstein. Near-linear-time deterministic plane Steiner spanners for well-spaced point sets. *Comput. Geom.*, 49:8–16, 2015. doi:10.1016/J.COMGEO.2015.04.005.

7 Prosenjit Bose, Joachim Gudmundsson, and Michiel H. M. Smid. Constructing plane spanners of bounded degree and low weight. *Algorithmica*, 42(3-4):249–264, 2005. doi:10.1007/S00453-005-1168-8.

8 Prosenjit Bose and Michiel H. M. Smid. On plane geometric spanners: A survey and open problems. *Comput. Geom.*, 46(7):818–830, 2013. doi:10.1016/J.COMGEO.2013.04.002.

9 T.-H. Hubert Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. *ACM Trans. Algorithms*, 12(4):55:1–55:22, 2016. doi:10.1145/2915183.

10 T.-H. Hubert Chan, Mingfei Li, Li Ning, and Shay Solomon. New doubling spanners: Better and simpler. *SIAM J. Comput.*, 44(1):37–53, 2015. doi:10.1137/130930984.

11 Kenneth L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th Annual ACM Symposium on Theory of Computing, STOC*, pages 56–65. ACM, 1987.

12 Vincent Cohen-Addad, Arnold Filtser, Philip N. Klein, and Hung Le. On light spanners, low-treewidth embeddings and efficient traversing in minor-free graphs. In *Proc. 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 589–600. IEEE, 2020. doi:10.1109/FOCS46700.2020.00061.

13 Sarita de Berg, Tim Ophelders, Irene Parada, Frank Staals, and Jules Wulms. The complexity of geodesic spanners using steiner points, 2024. arXiv:2402.12110, doi:10.48550/arXiv.2402.12110.

14 Sarita de Berg, Marc van Kreveld, and Frank Staals. The complexity of geodesic spanners. *CoRR*, abs/2303.02997, 2023. URL: https://arxiv.org/abs/2303.02997, doi:10.48550/arXiv.2303.02997.

15 Sarita de Berg, Marc J. van Kreveld, and Frank Staals. The complexity of geodesic spanners. In *Proc. 39th International Symposium on Computational Geometry, SoCG*, volume 258 of *LIPIcs*, pages 16:1–16:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.SOCG.2023.16.

16 Yefim Dinitz, Michael Elkin, and Shay Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 519–528, 2008. doi:10.1109/FOCS.2008.24.

17 Michael Elkin and Shay Solomon. Narrow-shallow-low-light trees with and without Steiner points. *SIAM J. Discret. Math.*, 25(1):181–210, 2011. doi:10.1137/090776147.

18 Michael Elkin and Shay Solomon. Optimal Euclidean spanners: Really short, thin, and lanky. *J. ACM*, 62(5):35:1–35:45, 2015. doi:10.1145/2819008.

19 Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient regression in metric spaces via approximate Lipschitz extension. *IEEE Trans. Inf. Theory*, 63(8):4838–4849, 2017. doi:10.1109/TIT.2017.2713820.

20 Lee-Ad Gottlieb and Liam Roditty. An optimal dynamic spanner for doubling metric spaces. In *Proc. 16th Annual European Symposium on Algorithms, ESA*, volume 5193 of *LNCS*, pages 478–489, 2008. doi:10.1007/978-3-540-87744-8_40.

21 Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006. doi:10.1137/S0097539704446281.

22 Hung Le and Shay Solomon. Truly optimal Euclidean spanners. In *Proc. 60th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1078–1100. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00069.

23 Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *Proc. 13th Annual ACM Symposium on the Theory of Computing, STOC*, pages 186–195. ACM, 1998. doi:10.1145/276698.276734.

24 Joseph S. B. Mitchell and Wolfgang Mulzer. Proximity algorithms. In *Handbook of Discrete and Computational Geometry (3rd Edition)*, chapter 32, pages 849–874. Chapman & Hall/CRC, 2017.

25 Giri Narasimhan and Michiel H. M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

26 Jan Remy, Reto Spöhel, and Andreas Weißl. On Euclidean vehicle routing with allocation. *Comput. Geom.*, 43(4):357–376, 2010. doi:10.1016/J.COMGEO.2008.12.009.