The Complexity of Geodesic Spanners using Steiner Points

Sarita de Berg[1], Tim Ophelders[1,2], Irene Parada[3], Frank Staals[1], and Jules Wulms[2]

1 **Department of Information and Computing Sciences, Utrecht University, the Netherlands**
`S.deBerg@uu.nl, T.A.E.Ophelders@uu.nl, F.Staals@uu.nl`

2 **Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands**
`J.H.H.M.Wulms@tue.nl`

3 **Department of Mathematics, Universitat Politècnica de Catalunya, Spain**
`Irene.Parada@upc.edu`

Abstract

A geometric t-spanner 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 in the spanner 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 total spanner complexity is a recently-introduced measure of how compact a spanner is. 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 on 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.

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 then apply it to obtain a $2\sqrt{2}t$-spanner in a simple polygon or a $6t$-spanner a in polygonal domain with total complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$.

Related Version A full version of the paper is available at `https://arxiv.org/abs/2402.12110`.

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 [10], vehicle routing [7, 14], and constructing TSP tours [5], 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 [12]. 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 [13].

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

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

this environment is modeled by a polygon P with m vertices, and possibly containing holes. The distance between a pair of 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 geodesic $(\sqrt{10}+\varepsilon)$-spanner of size $O(n \log^2 n)$ when P is a simple polygon, and a geodesic $(5+\varepsilon)$-spanner of size $O(n\sqrt{h}\log^2 n)$ when the polygon has $h > 1$ holes. Abam, de Berg, and Seraji [2] even obtained 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 [8], 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. The above spanners of [1, 2] all have worst-case complexity $\Omega(mn)$, hence they present an algorithm to construct a $2\sqrt{2}t$-spanner in a simple polygon or a $6t$-spanner in a polygon with holes with complexity $O(mn^{1/t} + n\log^2 n)$, for any integer $t > 1$.[1] These complexity bounds are still relatively high. De Berg, van Kreveld, and Staals [8] 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 (i.e. $d_{\mathcal{G}}(p,q) \leq td(p,q)$), 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 [4, 9] and size [11] 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 [6] 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.

Our contributions. Surprisingly, we show that in this setting Steiner points have only limited utility. In the full version, 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 an $\Omega(mn/k)$ and $\Omega(mn^{1/(1+t)}/k^{1/(1+t)})$ lower bound on the complexity of a $(3-\varepsilon)$- and a $(t-\varepsilon)$-spanner, respectively. 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

[1] De Berg, van Kreveld, and Staals [8] claim that the refinement by Abam, de Berg, and Seraji [2] can be applied to obtain a $(2t+\varepsilon)$-spanner of the same complexity (increased by a constant factor dependent on ε). However, some details of how this refinement influences the complexity are missing.

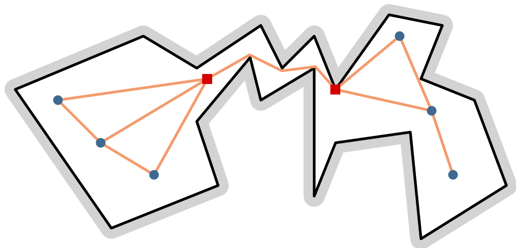

Figure 1 A spanner in a simple polygon that uses two Steiner points (red squares). By adding the two Steiner points, we no longer need multiple links that pass through the middle section of P.

leaves of the tree. In Section 2, we show that in this setting we can efficiently construct a spanner whose complexity is relatively close to optimal. In particular, our algorithm constructs a $2t$-spanner of complexity $O(mn^{1/t}/k^{1/t} + n\log(n/k))$. A slight 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 simple polygons. In particular, in Section 3, we combine some of the techniques developed by de Berg, van Kreveld, and Staals [8] and the Steiner spanner for a forest to build a $2\sqrt{2}t$-spanner of complexity $O(mn^{1/t}(\log k)^{1+1/t}/k^{1/t} + n\log^2 n)$. The main challenge here is to argue that the links used still have low complexity, even when they are now embedded in the polygon. For $k = O(1)$ our spanner thus matches the result of de Berg, van Kreveld, and Staals [8]. In the full version, we extend these results to polygonal domains, where we get a $6t$-spanner of similar complexity. Omitted proofs are contained in the full version.

2 Steiner spanners for trees

In this section, we consider spanners on an edge-weighted rooted tree T. We allow only positive weights. 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. In the full version, we prove several lower bounds on the complexity of any spanner that uses k Steiner points. Among these is a general lower bound of $\Omega(mn^{1/(t+1)}/k^{1/(t+1)})$ for any $(t - \varepsilon)$-spanner. The goal in this section is to construct a spanner of complexity close to this lower bound. We denote by $T(v)$ the subtree of T rooted at vertex v. For an edge $e \in T$ with upper endpoint v_1 and lower endpoint v_2, we denote by $T(e) := T(v_2) \cup \{e\}$ the subtree of e rooted at v_1. The following lemma states that we can build a low complexity spanner for a tree (without using Steiner points).

▶ **Lemma 2.1** (de Berg et al. [8]). *For any integer $t \geq 1$, we can build a $2t$-spanner for 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. To construct a Steiner spanner $\mathcal{G}$, 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 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 by color i. Note that after this step there are no uncolored vertices that have an incident descendant edge that is colored. In the next step, we color the remaining uncolored edges and vertices. These edges and their (possibly already colored) upper endpoints are colored using a bottom-up approach. 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 place a Steiner point s_i at the root of tree T_i formed by all edges and vertices of color i for $i \in \{1, \dots, k\}$. Observe that it may happen that more than one Steiner point is assigned to some vertex. Slightly abusing our notation, we denote the vertex that the Steiner point s_i is placed at by s_i as well.

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 2(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 2(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.

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 2.1. 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 $\mathcal{G}$, 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 2.2.** *An edge can have at most two colors.*

Proof. First of all, 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 the contradiction $p_\ell \in S_j$. ◀

▶ **Lemma 2.3.** *The sets S'_i and trees T'_i adhere to the four stated properties.*

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

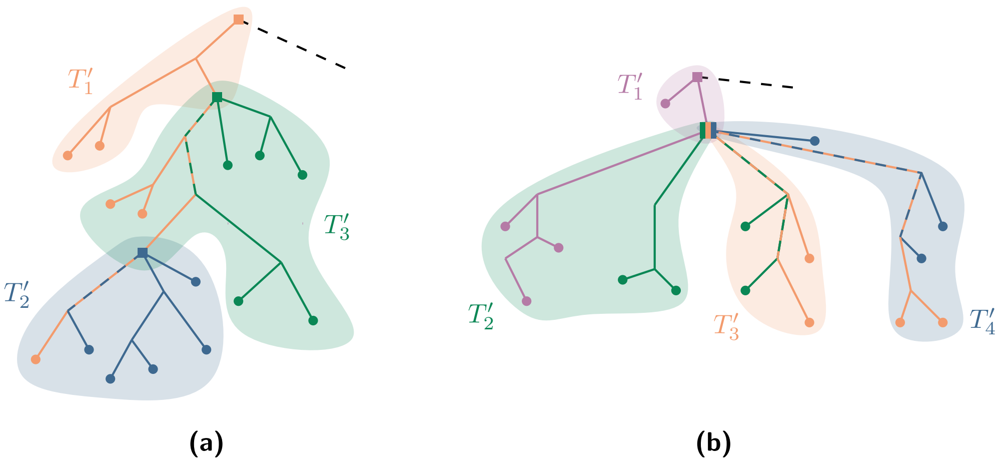

Figure 2 The tree T_i is the subtree whose edges and vertices have color i. A Steiner point (square) is placed at the root of T_i. The shaded areas show the trees T_i'. The examples show the case when the Steiner points are **(a)** at different vertices or **(b)** share a vertex.

▶ **Theorem 2.4.** *Let T be a tree with n leaves and m vertices. For any integer $t \geq 1$, we can build a $2t$-spanner $\mathcal{G}$ for T 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.*

Proof sketch. Let n_i and m_i denote the number of leaves and vertices in a subproblem T_i'. Properties 1 and 4 imply that $n_i = O(n/k)$, and property 3 implies that $\sum_i m_i = O(m)$. Lemma 2.1 then implies the size and complexity of $\mathcal{G}$. To bound the spanning ratio, consider the path $\pi(p,q)$ between two sites p, q. Properties 2 and 3 imply that this path exits a subtree T_i' and enters another subtree only at Steiner points. As within each subtree there is a $2t$-spanner, this is also the spanning ratio of $\mathcal{G}$. ◀

The output size K is either the size or complexity of $\mathcal{G}$, depending on whether we report the edges implicitly or explicitly. In the full version, we extend the result to a forest of trees.

3 Steiner spanners in simple polygons

We consider the problem of computing a t-spanner using k Steiner points for n point sites in a simple polygon P with m vertices. To obtain a low-complexity spanner we combine ideas from [2] and [8] with the forest spanner of Section 2.

We partition the polygon P recursively 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 shortest path tree SPT_λ. This shortest path tree includes all sites in S and vertices of P. The segment λ is split into multiple edges at the projections of the sites. See Figure 3 for an example. The tree is rooted at the lower endpoint of λ. Observe that there are $O(m+n)$ vertices in SPT_λ.

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 $SPT_{i,j}$. 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 $\mathcal{F}$. For this forest we construct a $2t$-spanner $\mathcal{G}_\mathcal{F}$. A Steiner point in $\mathcal{G}_\mathcal{F}$ corresponds to either a vertex of P or a point on λ. Let $\mathcal{S}$ denote

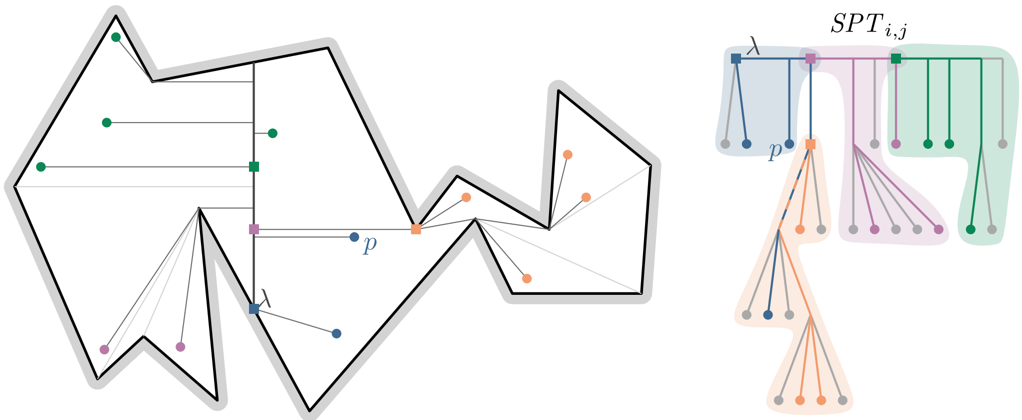

Figure 3 The shortest path tree of λ in P' and its $SPT_{i,j}$. The grey nodes and edges are not included in $SPT_{i,j}$, but can be assigned to a T_i' as indicated by the colored backgrounds. The squares show the Steiner points in $SPT_{i,j}$ and P'. The sites in P' are colored as the trees T_i'.

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.

To bound the complexity of the links in $\mathcal{G}$, we show that any path in a subtree T_i' as defined in Section 2 uses vertices of P in that subtree only. This implies that the bound on the complexity of the forest spanner also holds for the complexity of the links in the polygon. As the number of sites and vertices in $\mathcal{F}$ is increased by a factor $O(\log n)$ compared to n and m, we obtain the following theorem.

▶ **Theorem 3.1.** *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 $k \in \{1, \dots, n\}$, we can build a geodesic $2\sqrt{2}t$-spanner with at most k Steiner points, 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 $O(n \log^2 n + m \log n + K)$ time, where K is the output size.*

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.

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.

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.

4 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.

5 Glencora Borradaile and David Eppstein. Near-linear-time deterministic plane Steiner spanners for well-spaced point sets. *Comput. Geom.*, 49:8–16, 2015.

6 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.

7 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.

8 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.

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

10 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.

11 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.

12 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.

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

14 Jan Remy, Reto Spöhel, and Andreas Weißl. On Euclidean vehicle routing with allocation. *Comput. Geom.*, 43(4):357–376, 2010.