

EUCLIDEAN STEINER SHALLOW-LIGHT TREES*

Shay Solomon[†]

ABSTRACT. A spanning tree that simultaneously approximates a shortest-path tree and a minimum spanning tree is called a *shallow-light tree* (shortly, SLT). More specifically, an (α, β) -SLT of a weighted undirected graph $G = (V, E, w)$ with respect to a designated vertex $rt \in V$ is a spanning tree of G with: (1) *root-stretch* α – it preserves all distances between rt and the other vertices up to a factor of α , and (2) *lightness* β – it has weight at most β times the weight of a minimum spanning tree $MST(G)$ of G .

Tight tradeoffs between the parameters of SLTs were established by Awerbuch et al. in PODC'90 and by Khuller et al. in SODA'93. They showed that for any $\epsilon > 0$, any graph admits a $(1 + \epsilon, O(\frac{1}{\epsilon}))$ -SLT with respect to any root vertex, and complemented this result with a matching lower bound.

Khuller et al. asked if the upper bound $\beta = O(\frac{1}{\epsilon})$ on the lightness of SLTs can be improved in Euclidean spaces. In FOCS'11 Elkin and this author gave a negative answer to this question, showing a lower bound of $\beta = \Omega(\frac{1}{\epsilon})$ that applies to 2-dimensional Euclidean spaces.

In this paper we show that *Steiner points lead to a quadratic improvement in Euclidean SLTs*, by presenting a construction of Euclidean Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLTs in arbitrary 2-dimensional Euclidean spaces. The lightness bound $\beta = O(\sqrt{\frac{1}{\epsilon}})$ of our construction is optimal up to a constant. The runtime of our construction, and thus the number of Steiner points used, are bounded by $O(n)$.

1 Introduction

1.1 Background

Consider a weighted undirected n -vertex graph $G = (V, E, w)$, $w : E \rightarrow \mathbb{R}^+$. A *minimum spanning tree* (shortly, MST) of G is a spanning tree $T = (V, H, w)$ of G of minimum weight $\omega(T) = \sum_{e \in H} \omega(e)$. A *shortest-path tree* (shortly, SPT) of G with respect to a root vertex $rt \in V$ is a spanning tree $T = (V, H, w)$ that preserves all distances from rt , i.e., for every vertex $v \in V$, the distance $d_T(rt, v)$ between rt and v in T is equal to their distance $d_G(rt, v)$ in G . The MST and the SPT are among the most fundamental graph constructs, and have been extremely well studied over the years.

*This work is supported by the Koshland Center for basic Research.

[†]Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel, shay.solomon@weizmann.ac.il

A spanning tree that simultaneously approximates a shortest-path tree and a minimum spanning tree is called a *shallow-light tree* (shortly, SLT). This notion was introduced in [5, 6, 29] (and goes by the name of *LAST* in [29]). For a pair of parameters $\alpha, \beta \geq 1$, an (α, β) -SLT of a weighted undirected graph $G = (V, E, w)$ with respect to a designated vertex $rt \in V$ is a spanning tree T of G with:

1. *root-stretch* α – it preserves all distances between rt and the other vertices up to a factor of α .
2. *lightness* β – it has weight at most β times the weight of a minimum spanning tree $MST(G)$ of G .

Awerbuch et al. [5, 6] and Khuller et al. [29] independently showed that for every $\epsilon > 0$, a $(1 + \epsilon, O(\frac{1}{\epsilon}))$ -SLT exists for every graph G . Moreover, this tradeoff was shown to be tight in [29]. SLTs were found useful for numerous data gathering and dissemination tasks in overlay networks [10, 44, 34], in the message-passing model of distributed computing [5, 6], and in wireless and sensor networks [45, 9, 18, 36, 35, 43]. Moreover, SLTs find applications in routing [3, 42, 28, 46] and in network and VLSI-circuit design [15, 16, 17, 41]. In addition, SLTs are embedded within various related structures, such as *light approximate routing trees* [46], *shallow-low-light trees* [19, 20], *light spanners* [6, 40], and others [41, 36, 35].

The plethora of both theoretical and practical applications of SLTs, only some of which are listed above, testifies that SLTs constitute a fundamental graph structure of independent interest.

A natural arising question, proposed by Khuller et al. [29] in SODA'93, is whether the upper bound $\beta = O(\frac{1}{\epsilon})$ on the lightness of SLTs can be improved in Euclidean spaces. In FOCS'11 [21] Elkin and this author provided a negative answer to this question, showing a lower bound of $\beta = \Omega(\frac{1}{\epsilon})$, for $\epsilon = \Omega(\frac{1}{n})$, that applies to 2-dimensional Euclidean spaces. Specifically, the lower bound of [21] applies to a set C_n of n points evenly spaced along the boundary of a circle, and so it holds for any choice of root vertex.

Euclidean Steiner Trees. A *Steiner tree* for a set P of n points in \mathbb{R}^2 is a tree $T = (P', H, \|\cdot\|)$ spanning a superset $P' \supseteq P$ of points, where H is a subset of the $\binom{|P'|}{2}$ segments connecting pairs of points from P' , and the weight $\omega(e)$ of each edge $e = (p, q) \in H$ is the Euclidean distance $\|p, q\|$ between its endpoints. The points in $P' \setminus P$, called *Steiner points*, may help to improve the quality of the tree.

A *Steiner minimum tree* (shortly, SMT) for P is a Steiner tree $T = (P', H, \|\cdot\|)$ of minimum weight $\omega(T) = \sum_{e \in H} \omega(e)$. The *Steiner ratio* is defined as the smallest possible ratio between the weight of the SMT and that of the MST. In any metric, the Steiner ratio is at least $\frac{1}{2}$ (by the triangle inequality) and at most 1 (by definition). In 2-dimensional Euclidean spaces, the Steiner ratio is between ≈ 0.824 and $\frac{\sqrt{3}}{2} \approx 0.866$, and the famous “Gilbert-Pollak Conjecture” is that the upper bound $\frac{\sqrt{3}}{2}$ is tight [23, 14, 27].

For a pair of parameters $\alpha, \beta \geq 1$, a *Steiner* (α, β) -SLT for P with respect to a designated vertex $rt \in P$ is a Steiner tree T of P with:

1. *root-stretch* α – it preserves all distances between rt and the other points of P up to

a factor of α .

2. *lightness* β – it has weight at most β times the weight of a *Steiner minimum tree* $SMT(P)$ of P . (Notice that the lightness here is defined with respect to $SMT(P)$ rather than $MST(P)$.)

As mentioned, there is a lower bound of $\beta = \Omega(\frac{1}{\epsilon})$ on the lightness of Euclidean *spanning* SLTs, which applies to a set C_n of n evenly spaced points on the boundary of a circle. For Euclidean *Steiner* trees, there is a weaker lower bound of $\beta = \Omega(\sqrt{\frac{1}{\epsilon}})$, for $\epsilon = \Omega(\frac{1}{n^2})$, which applies to the same point set C_n [21]. (See Appendix A for details.)

1.2 Our Results

In this paper we study the impact of *Steiner points* in the context of Euclidean SLTs. Specifically, we show that for every $\epsilon > 0$, any point set $P \in \mathbb{R}^2$ and every designated point $rt \in P$, there is a Euclidean *Steiner* $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLT with respect to rt . Our bound $\beta = O(\sqrt{\frac{1}{\epsilon}})$ on the lightness matches (up to a constant factor hidden by the O -notation) the lower bound of [21]. Recall that another lower bound of [21] shows that the lightness of Euclidean *spanning* SLTs with the same root-stretch is $\Omega(\frac{1}{\epsilon})$, and so we conclude that *Steiner points lead to a quadratic improvement in Euclidean SLTs*.

Our construction contains $O(n)$ Steiner points. Also, it can be implemented in $O(n)$ time in the standard real-RAM model, which is the main model of computability in Computational Geometry. In this model memory cells are allowed to store arbitrary real numbers (such as coordinates of points), and all basic mathematical functions, including the floor function, can be computed in constant time. (Refer to [1] for a more detailed description of the real-RAM model.)

We did not make an effort to optimize the leading constant in the lightness bound $O(\sqrt{\frac{1}{\epsilon}})$. Since the SMT and the MST differ by a constant, re-defining the lightness of Steiner SLTs with respect to the MST will make no difference.

1.3 Do Steiner Points Really Help (in Euclidean Trees)?

The impact of Steiner points on *Euclidean trees* has been very well studied. This line of research might lead to the impression that Steiner points do not help beyond constants, in which case finding/achieving the optimal constants becomes the ultimate goal.

The SMT problem is a good example: It is known that Steiner points can reduce the weight of the MST by a small constant factor (between ≈ 0.824 and ≈ 0.866), and determining the exact constant is a central open question coinciding with the aforementioned Gilbert-Pollak Conjecture.

Moreover, there are settings in which Steiner points are completely useless. For example, in a recent COCOON'12 paper [11], it was shown that Steiner points cannot help

at all in reducing the weight of the MST (1) in a natural *budget allocation* model, or (2) under the Euclidean square root metric.

A similar situation is with *maximum-stretch trees*. While the maximum-stretch of a Euclidean MST is $O(n)$, any Euclidean Steiner tree for the point set C_n from Section 1.1 has maximum-stretch $\Omega(n)$ [4].

Yet another example occurs in the context of *low-light trees*, which combine small lightness with small depth [19, 20]. It is known that Steiner points do not help in this context either: Any Euclidean Steiner tree T can be converted into a spanning tree with the same (up to constants) lightness and depth as those of T [20]. See Section 1.5 for more examples of this “incompetence” of Steiner points.

In sharp contrast to previous results, Euclidean SLTs can be drastically improved using Steiner points!

1.4 Proof Overview

Our strategy for obtaining optimal Euclidean Steiner SLTs is to identify a *core example*, i.e., a simple example that manages to encapsulate the inherent complexity of the problem. Specifically, our core example is a set of evenly spaced points lying on the base of an isosceles triangle with apex angle $\Theta(\sqrt{\epsilon})$, plus another point at the apex of the triangle designated as the root of the SLT.

Constructing a Steiner $(1 + \epsilon, O(1))$ -SLT for the core example won’t be too difficult (see Section 2.1).

We next sketch a reduction from the problem of constructing Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLTs in arbitrary 2-dimensional Euclidean spaces to that of constructing a Steiner $(1 + \epsilon, O(1))$ -SLT for the core example.

The starting point is the construction of SLTs for general graphs due to Awerbuch et al. [5, 6], adapted to 2-dimensional Euclidean spaces. This construction starts by building an “MST-TOUR” $L = (v_1 = rt, v_2, \dots, v_n)$ for the input Euclidean point set P , which is a Hamiltonian path of weight at most $2 \cdot \omega(MST(P))$. While L is light, its root-stretch may be unbounded. To control the root-stretch, the idea is to identify a set $B = \{b_1 = rt, \dots, b_k\}$ of special points (consecutive along L), called *break-points*, and connect each break-point b_i to rt via a direct edge $e_i = (rt, b_i)$, $i = 2, \dots, k$.

The returned SLT is the SPT rooted at rt over the union of the edges in L and the $k - 1$ edges e_2, \dots, e_k .

Identifying the break-points is the tricky part. Fix an arbitrary parameter $\theta \ll 1$; having assigned $b_1 = rt, \dots, b_i$, the next break-point b_{i+1} is the first point after b_i along L such that $d_L(b_i, b_{i+1}) > \theta \cdot \|rt, b_{i+1}\|$. This means that: (i) The *path distance* $d_L(b_i, v)$ between any point $v \notin B$ and its preceding break-point b_i along L is at most $\theta \cdot \|rt, v\|$, so the concatenation of edge e_i and the subpath of L between b_i and v gives a $(1 + 2\theta)$ -*path* between rt and v (i.e., a path of weight at most $(1 + 2\theta) \cdot \|rt, v\|$). (ii) The weight of each edge $e_{i+1} = (rt, b_{i+1})$ can be “charged” to the path distance $d_L(b_i, b_{i+1})$ between b_i and b_{i+1} (up to a slack of $\frac{1}{\theta}$), thus the total weight of all edges e_2, \dots, e_k is at most $\frac{1}{\theta} \cdot L$, and so the

lightness is $O(\frac{1}{\theta})$.

We apply the above construction with $\theta = O(\sqrt{\epsilon})$ to get a $(1 + O(\sqrt{\epsilon}), O(\sqrt{\frac{1}{\epsilon}}))$ -SLT T^* . While the lightness bound is in check, the root-stretch is still too large. Denote by L_i the subpath of L between b_i and b_{i+1} , disregarding b_{i+1} , and let P_i be the set of points in P lying on that path. Define T_i as the spanning tree of the point set $P_i^+ = \{rt\} \cup P_i$ obtained from the union of edge $e_i = (rt, b_i)$ and path L_i , and note that the “role” of T_i is to “take care” of distances between rt and all points $v \in P_i$. A simple observation (but not trivial; see Lemma 4) that we will use next is that the weight of L_i is $O(\sqrt{\epsilon}) \cdot \omega(e_i)$.

Next, to decrease the stretch between rt and the points in P_i , for $1 \leq i \leq k$, we replace each tree $T_i = e_i \cup L_i$ by a Steiner tree for P_i^+ rooted at rt , with root-stretch $1 + O(\epsilon)$ and weight $O(\omega(T_i))$; this will guarantee low root-stretch while increasing the lightness by only a constant. As $\omega(e_i) \leq \omega(MST(P_i^+)) \leq \omega(T_i) \leq (1 + O(\sqrt{\epsilon})) \cdot \omega(e_i)$, we actually look for a Steiner $(1 + O(\epsilon), O(1))$ -SLT for P_i^+ rooted at rt :

1. The above observation implies that all points in P_i are at distance roughly $\omega(e_i)$ from rt (up to an additive slack of $O(\sqrt{\epsilon}) \cdot \omega(e_i)$). Consider the largest isosceles triangle \triangle with apex rt and apex angle $\alpha = \Theta(\sqrt{\epsilon})$ that do not have any point from P_i in its interior, such that the straight line connecting rt and an arbitrary point w from P_i is a bisector of α . (It is instructive to take w to be the point in P_i closest to rt , but any point will do.) It is easy to see that the triangle base has length $\Theta(\sqrt{\epsilon}) \cdot \omega(e_i)$. (See Figure 1.(i) for an illustration.) Moreover, the straight line connecting rt and any point $v \in P_i$ intersects the triangle base, at some point denoted \tilde{v} . (See Figures 1.(ii) and 2.(i) for an illustration.)
2. We distribute $O(\sqrt{\frac{1}{\epsilon}})$ evenly spaced Steiner points on the triangle base, so that the distance between any two consecutive Steiner points is $O(\epsilon) \cdot \omega(e_i)$. For each $v \in P_i$, let v' be the Steiner point on the triangle base closest to \tilde{v} . Clearly, the path (rt, v', v) is a $(1 + O(\epsilon))$ -path between rt and v . This means that if we add the two edges (rt, v') and (v', v) for each $v \in P_i$, the root-stretch will be $1 + O(\epsilon)$. (See Figure 2 for an illustration.)
3. Alas, we cannot add all edges $\bigcup_{v \in P_i} \{(rt, v'), (v', v)\}$, as the weight will exceed $O(\omega(T_i)) = O(\omega(e_i))$.

Note that any edge (v', v) , for $v \in P_i$, has weight $O(\sqrt{\epsilon}) \cdot \omega(e_i)$, and so we can safely add $O(\sqrt{\frac{1}{\epsilon}})$ such edges. Since the weight of L_i is $O(\sqrt{\epsilon}) \cdot \omega(e_i)$, it is easy to see that P_i can be “covered” via an ϵ -net $N_i \subset P_i$ of size $O(\sqrt{\frac{1}{\epsilon}})$, in the sense that each point $v \in P_i$ will have a nearby point $p(v)$ from the ϵ -net N_i satisfying $d_L(v, p(v)) \leq \epsilon \cdot \omega(e_i)$. This means (but requires proof of course) that it suffices to add edges (v', v) just for the $O(\sqrt{\frac{1}{\epsilon}})$ points v from the ϵ -net N_i .

It is left to consider edges (rt, v') , for $v \in P_i$. Instead of adding such edges, we will build a Steiner $(1 + \epsilon, O(1))$ -SLT rooted at rt , which spans all the $O(\sqrt{\frac{1}{\epsilon}})$ evenly spaced

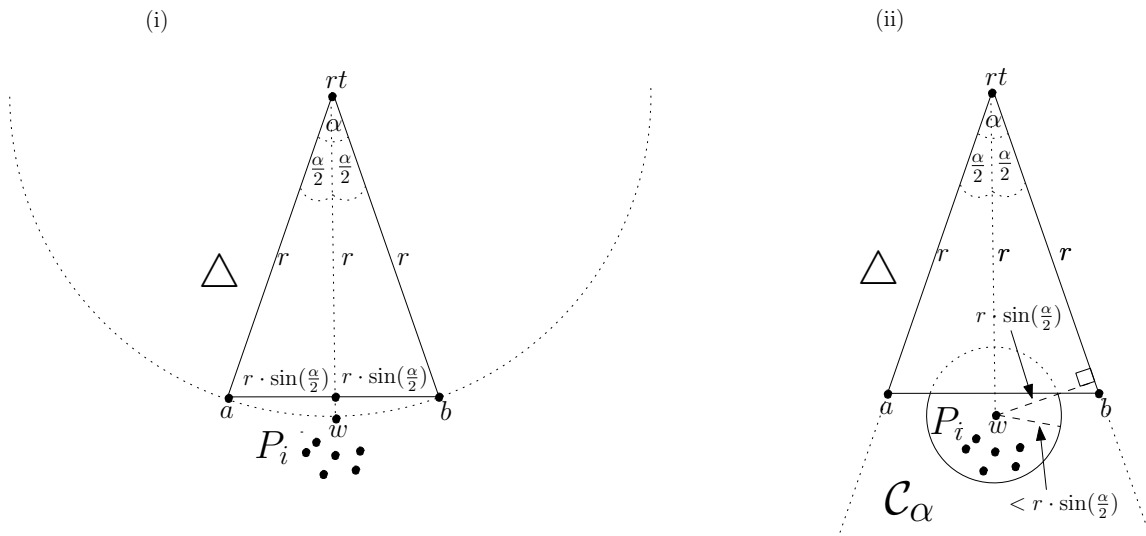


Figure 1: (i) An illustration of the isosceles triangle $\Delta = (rt, a, b)$ of side length $r := \|rt, w\|$ separating rt from P_i . No point of P_i may lie in the interior of the circle centered at rt with radius r . (ii) An illustration of the cone C_α defined by the two rays coinciding with the sides of triangle Δ . The Euclidean distance between w and the boundaries of cone C_α is at least $r \cdot \sin(\frac{\alpha}{2})$. Since all points of P_i belong to the circle centered at w with radius $2c\sqrt{\epsilon} \cdot \omega(e_i) < r \cdot \sin(\frac{\alpha}{2})$, all of them must lie within cone C_α .

Steiner points lying on the triangle base. Note that this is exactly the aforementioned core example (that we know how to solve), thus completing the reduction.

In this way we get a Steiner tree rooted at rt for P , with root-stretch $1 + O(\epsilon)$ and lightness $O(\sqrt{\frac{1}{\epsilon}})$. We can reduce the stretch to $1 + \epsilon$ by scaling the lightness up by some constant. The number of Steiner points may be as large as $O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$, but it can be reduced to $O(n)$ via simple adjustments.

1.5 Related Work

The impact of Steiner points on non-geometric trees was also extensively studied. A *Steiner tree* for an arbitrary metric $M = (V, \delta)$, is a tree $T = (V', H, \omega')$, with $V \subseteq V'$ and $\omega' : H \rightarrow \mathbb{R}^+$, that *dominates* the metric M , i.e., for every pair of original points $u, v \in V$, $d_T(u, v) \geq \delta(u, v)$. Even under this relaxed notion of Steiner trees, it is known that Steiner points almost always do not help much. A prime example is in the context of *low average-stretch trees*. Bartal and Fakcharoenphol et al. [7, 8, 22] devised constructions of Steiner trees for arbitrary metrics with the optimal average-stretch $O(\log n)$, which was a significant improvement over the previous constructions of spanning trees [2]. However, Konjevod et al. [32] and Gupta [25] demonstrated that the same bounds (up to constants) as those of [7, 8, 22] can be obtained without Steiner points. In fact, Gupta [25] showed a

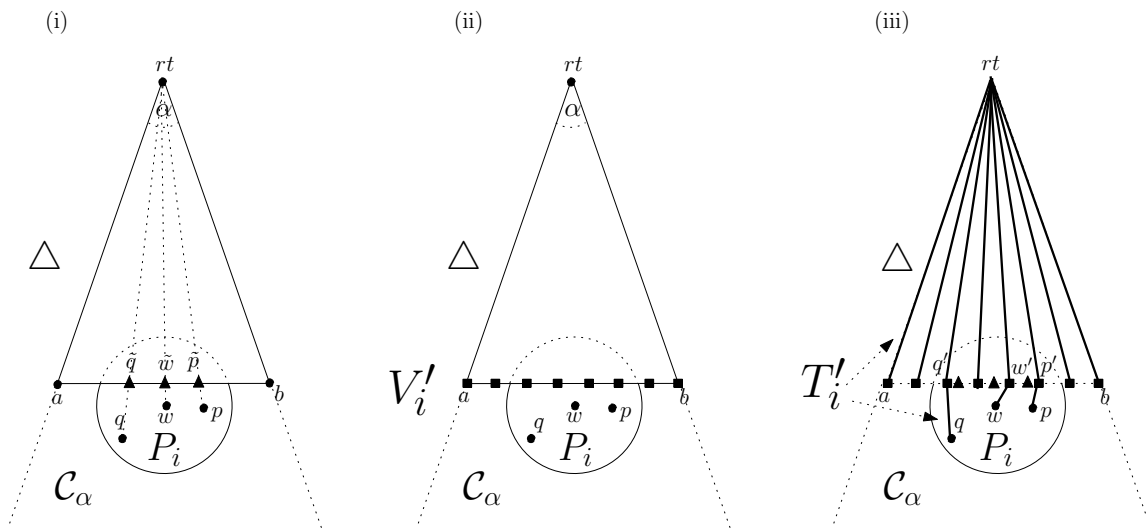


Figure 2: (i) An illustration of the straight lines connecting rt with the points of P_i , all of which must intersect the triangle base. Here $P_i = \{w, p, q\}$, and the lines connecting rt with w, q and p intersect the triangle base at points \tilde{w}, \tilde{q} and \tilde{p} , respectively, each depicted by a small filled triangle. (ii) An illustration of the set V_i' of evenly spaced points along the boundary of the triangle base, each depicted by a small filled square. (iii) An illustration of the Steiner tree T_i' for P_i^+ with root-stretch $1 + O(\epsilon)$, obtained from the union of the star S_i and the edge set E_i' , where $E(S_i) = \{(rt, v') \mid v' \in V_i'\}$ and $E_i' = \{(v', v) \mid v \in P_i\}$. Here $E(S_i) \supset \{(rt, w'), (rt, q'), (rt, p')\}$ and $E_i' = \{(w', w), (q', q), (p', p)\}$.

more general result, specifically, one can *prune* any Steiner tree $T = (V', H, \omega')$ from Steiner points to get a tree $T^* = (V, E, \omega)$ that preserves all pairwise distances of T (and has the same weight) up to constants. The *low-light trees* of [19, 20] mentioned in Section 1.3 apply in fact to arbitrary metrics, and also there Steiner points do not help beyond constants. We refer to [21] for more examples where Steiner points do not help much.

The only “tree setting” we are aware of, where Steiner points do help significantly, is in the context of SLTs. Specifically, it was shown in [21] that for any metric $M = (V, \delta)$ (or any graph) and every point $rt \in V$, there exists a $(1 + \epsilon, O(\log \frac{1}{\epsilon}))$ -SLT with respect to rt , yielding an exponential improvement in the lightness. However, this result is not applicable to Euclidean spaces. Indeed, even if the input metric M is a Euclidean 2-dimensional space, the SLTs of [21] will use Steiner points that do not belong to \mathbb{R}^d ! This weak spot in the work of [21] was the trigger to our current study of Euclidean Steiner SLTs.

Approximation algorithms for spanning and Steiner SLTs were studied in [37, 13, 33, 38, 26]; online approximation algorithms for SLTs were given in [24]. Heuristics for finding SLTs were developed in [31].

Another closely related notion is *shallow-low-light trees* [19, 20]. In addition to small root-stretch and lightness, these trees have small depth. Similarly to the SLTs of [5, 6, 29], the shallow-low-light trees of [19, 20] exhibit an inverse-linear tradeoff of $1 + \epsilon$ versus $\Omega(\frac{1}{\epsilon})$ between the root-stretch and lightness.

1.6 Structure of the Paper

In Section 2 we present our construction of Euclidean Steiner SLTs, and analyze its root-stretch and lightness. In Section 3 we analyze the running time of the construction. Finally, in Section 4 we give our conclusions and discuss some directions for future work.

1.7 Preliminaries

Let $T = (T, rt)$ be either a Euclidean spanning or Steiner tree of a point set $P \in \mathbb{R}^2$ rooted at some designated point rt . For a pair of points p, q in P , denote by $\|p, q\|$ and $d_T(p, q)$ their Euclidean distance and tree distance, respectively; clearly $d_T(p, q) \geq \|p, q\|$. The *stretch* between p and q in T is defined as $Str_T(p, q) = \frac{d_T(p, q)}{\|p, q\|}$. The *root-stretch* of (T, rt) is defined as $RtStr(T, rt) = \max \{Str_T(rt, p) \mid p \in P \setminus \{rt\}\}$. The weight $\omega(T)$ of tree T is the sum of all edge weights in it, i.e., $\omega(T) = \sum_{e \in T} \omega(e)$; similarly, the weight $\omega(\Pi) = \sum_{e \in \Pi} \omega(e)$ of path Π is the sum of all edge weights in it. We say that a path Π between a pair p, q of points in P is a t -path, for $t \geq 1$, if $\frac{\omega(\Pi)}{\|p, q\|} \leq t$. We denote by $\Psi(T) = \frac{\omega(T)}{\omega(MST(P))}$ the *lightness* of a Euclidean spanning or Steiner tree T of P . For a pair k, n of integers, $0 \leq k \leq n$, denote the sets $\{k, k+1, \dots, n\}$ and $\{1, 2, \dots, n\}$ by $[k, n]$ and $[n]$, respectively.

2 Euclidean Steiner SLTs

In this section we show that for any 2-dimensional Euclidean n -point set P , any point $rt \in P$, and any $\epsilon > 0$, there is a Euclidean Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLT \hat{T} for P with respect to rt .

Our construction of Steiner SLTs in 2-dimensional Euclidean spaces is obtained via a two-step strategy:

1. In Section 2.1 we identify a *core example* ϑ , i.e., a simple example that manages to encapsulate the inherent complexity of the problem. We then show how to construct a Steiner $(1 + \epsilon, O(1))$ -SLT for ϑ .
2. In Section 2.2 we provide a reduction from the problem of constructing Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLTs in arbitrary 2-dimensional Euclidean spaces to that of constructing a Steiner $(1 + \epsilon, O(1))$ -SLT for ϑ .

2.1 The Core Example

Our core example ϑ consists of a set V' of evenly spaced points lying on the base of an isosceles triangle with apex angle $\Theta(\sqrt{\epsilon})$, plus another point at the apex of the triangle designated as the root of the SLT.

We denote this triangle by $\Delta = (rt, a, b)$, where a and b are the endpoints of the triangle base, and rt is the apex of the triangle. Suppose without loss of generality that the side length of Δ equals 1, and let $\alpha := \sqrt{\epsilon}$ be the apex angle of triangle Δ (the smaller α is, the better root-stretch we get). Note that the length of the triangle base is given by $2 \sin(\frac{\alpha}{2}) \approx \alpha$ and the altitude of Δ is given by $\cos(\frac{\alpha}{2}) \approx 1 - \frac{\alpha^2}{8}$.

A Warm Up. We first build a Steiner $(1 + \epsilon, O(\log |V'|))$ -SLT T^{core} for the point set $\vartheta = \{rt\} \cup V'$.

The construction is carried out recursively. (See Figure 3.(a) for an illustration.)

Break the triangle base into two equal parts, each serving as the base of an isosceles triangle contained within Δ , with the same apex angle α but half the side length. This gives us two congruent triangles Δ_1 and Δ_2 , whose apexes rt_1 and rt_2 lie in the middle of the sides (rt, a) and (rt, b) of Δ , respectively.

Let V'_1 and V'_2 be the subsets of all points in V' belonging to the triangle bases of Δ_1 and Δ_2 , respectively. We connect rt to the apexes rt_1 and rt_2 of triangles Δ_1 and Δ_2 , respectively, with rt_1 and rt_2 serving as Steiner points in our Steiner SLT T^{core} . Next, we proceed recursively to building Steiner SLTs T_1^{core} and T_2^{core} for the point sets $\vartheta_1 = \{rt_1\} \cup V'_1$ and $\vartheta_2 = \{rt_2\} \cup V'_2$ “within” triangles Δ_1 and Δ_2 , respectively.

The recursion bottoms once we reach an isosceles triangle with only one point from V' lying on its base, in which case the SLT T^{core} contains a single edge connecting the triangle’s apex with that point.

Lightness analysis. Notice that $\omega(MST(\vartheta)) \approx \alpha + 1 - \frac{\alpha^2}{8} < 2$. Note also that the recursion

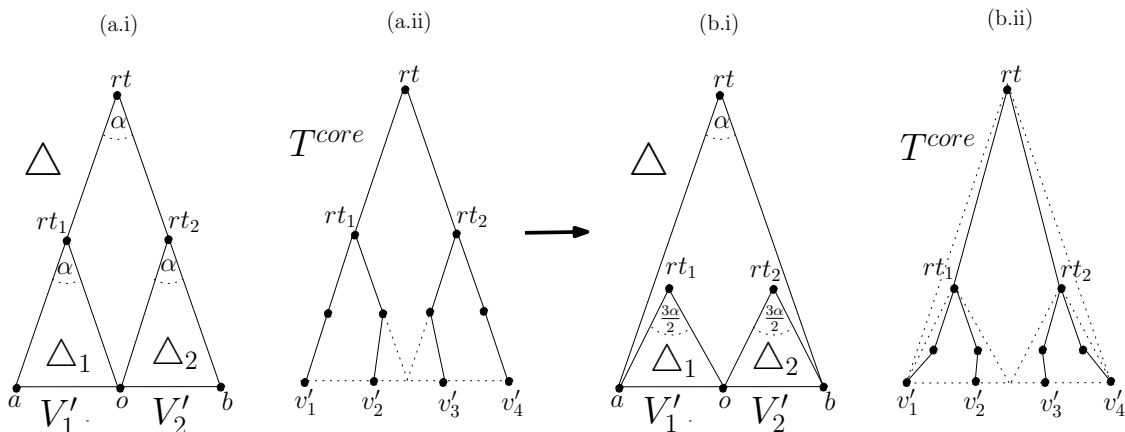


Figure 3: (a) An illustration of (a.i) triangle Δ and smaller triangles Δ_1 and Δ_2 contained within Δ , all with an apex angle of α ; (a.ii) the corresponding Steiner tree T^{core} for $V' = \{v'_1, v'_2, v'_3, v'_4\}$, with dotted lines drawing the borders of the underlying triangles (the borders of the 4 bottom-level triangles are not depicted in the figure). (b) A similar illustration, but here (b.i) the apex angle of Δ_1 and Δ_2 is $\frac{3\alpha}{2}$ rather than α ; (b.ii) the Steiner tree T^{core} has smaller (and, in general, constant) lightness.

bottoms after $\log |V'|$ levels. Consider the first level of the recursion, in which we add the two edges (rt, rt_1) and (rt, rt_2) to T^{core} . Since $\|rt, rt_1\| = \|rt, rt_2\| = \frac{1}{2}$, the total edge weights added in this level equals 1. While the number of edges grows geometrically with the recursion level, their weights decrease geometrically at the same rate. Thus the total edge weights added in each level of the recursion (except for the bottom level) equals 1. Also, it is easy to see that the side length of the triangles at the bottom level of the recursion is equal to $\frac{1}{|V'|}$, so the total edge weights added at the bottom level of the recursion is bounded by 1. It follows that the weight of T^{core} is at most $\log |V'| + 1$, thus the lightness is $O(\log |V'|)$.

Root-stretch analysis. Denote the point in the middle of the triangle base by o . Consider an arbitrary point w in V'_1 (without loss of generality), and note that $\|rt_1, w\| \leq \|rt_1, a\| = \|rt_1, o\| = \frac{1}{2}$. We also have $\|rt, rt_1\| = \frac{1}{2}$, yielding $\|rt, rt_1\| + \|rt_1, w\| \leq 1$, and so the weight of the path (rt, rt_1, w) is at most 1. On the other hand, the distance between rt and any point of V'_1 is at least $\|rt, o\| \approx 1 - \frac{\alpha^2}{8} = 1 - \frac{\epsilon}{8}$. This means that, when going from rt to any point of V'_1 , the additive slack incurred by taking a detour through rt_1 is at most $\frac{\epsilon}{8}$. Since we do not have direct edges from rt_1 to points in V'_1 , we will need to apply this argument recursively. It is easy to see that the additive slack incurred at each level of the recursion decreases geometrically, at the same rate as the lengths of the triangle sides decrease. Thus the total additive slack to all distances from rt can be bounded via the following sum $\frac{\epsilon}{8} \cdot (1 + \frac{1}{2} + \frac{1}{4} + \dots) \leq \frac{\epsilon}{4}$. Finally, recall that the minimum distance between rt and any point of V' is $\|rt, o\|$, which is bounded below by say $1 - \frac{\epsilon}{2}$. It follows that the root-stretch of T^{core} is at most $1 + \frac{\frac{\epsilon}{4}}{1 - \frac{\epsilon}{2}} < 1 + \epsilon$ (assuming $\epsilon < 1$).

Achieving Constant Lightness. We showed how to get a Steiner $(1 + \epsilon, O(\log |V'|))$ -SLT T^{core} for $\{rt\} \cup V'$. With minor adjustments, we can reduce the lightness bound to constant. (See Figure 3.(b).)

The aforementioned construction starts off with an isosceles triangle \triangle of apex angle α , and proceeds recursively with two isosceles triangles \triangle_1 and \triangle_2 built on the two halves of the triangle base, having apex angle α each. Hence, in all levels of the recursion the triangles preserve an apex angle of α . To reduce the lightness, we increase the triangles' apex angle by a factor of ξ in each recursive call, for an appropriate constant $\xi > 1$, so that in level i of the recursion the triangles' apex angle will be $\alpha \cdot \xi^i$.

In particular, taking $\xi = \frac{3}{2}$ does the job:

Lightness analysis. The total edge weights in each level of the recursion will decrease geometrically by a factor of roughly $\frac{2}{3}$, giving a total weight of at most $\frac{1}{\cos(\frac{\alpha}{2})} \cdot (\frac{4}{3} + \frac{8}{9} + \frac{16}{27} + \dots) \leq \frac{4}{\cos(\frac{\alpha}{2})}$, hence constant lightness.

Root-stretch analysis. The additive slack on the root-distances will no longer decrease by a factor of 2 at each level, but rather by a factor of roughly $\frac{4}{3}$. Consequently, the additive slack can be bounded via the following sum $\frac{\epsilon}{8} \cdot (1 + \frac{3}{4} + \frac{9}{16} + \dots) \leq \frac{\epsilon}{2}$. Similarly to before, the root-stretch will be at most $1 + \frac{\frac{\epsilon}{2}}{1 - \frac{\epsilon}{2}} < 1 + \epsilon$ (assuming $\epsilon < 1$).

The next observation follows immediately from the construction.

- Observation 1.**
1. T^{core} is a binary Steiner tree for $\{rt\} \cup V'$, with points in V' serving as its leaves.
 2. Each internal vertex in T^{core} has exactly two children, except for the leaves' parents which have a single child each. In particular, the total number of points in T^{core} is at most $|V'| - 1 + |V'| + |V'| = O(|V'|)$.
 3. This construction can be implemented within $O(|V'|)$ time in the obvious way.

2.2 A Reduction to the Core Example

In this section we construct a $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLT $\hat{T} = \hat{T}(P)$ for an arbitrary 2-dimensional Euclidean n -point set P . Our construction consists of two phases that we describe next. We remark that the construction from Section 2.1 (for the core example ϑ) will be used as a “black-box” in the second phase.

We assume throughout that $\epsilon < \frac{1}{4}$.

Phase 1: Spanning SLTs with Poor Root-Stretch.

Let T be an MST of P rooted at an arbitrary point $rt \in P$, and let D be an Euler tour of T starting at rt . For every $p \in P$, remove from D all occurrences of p except for the first one, and denote by $L = \{v_1 = rt, v_2, \dots, v_n\}$ the resulting Hamiltonian path of P . Observe that $\omega(L) \leq 2 \cdot \omega(T) = 2 \cdot \omega(MST(P))$. Also, L can be constructed in linear time, given T . Fix a parameter $\theta \ll 1$. The value of θ will determine the values of the root-stretch and lightness of the constructed SLT. We start with identifying a set of “break-points” $B = \{b_1, \dots, b_k\}$,

$B \subseteq P$. The first break-point b_1 is rt . The break-point b_{i+1} , $i \in [k-1]$, is the first vertex in L after b_i such that

$$d_L(b_i, b_{i+1}) > \theta \cdot \|rt, b_{i+1}\|. \quad (1)$$

The *path distance* between a pair u, v of points is defined as the distance $d_L(u, v)$ between them in L . Let S be the set of edges connecting rt with all other break-points, i.e., $S = \{e_i := (rt, b_i) : i \in [2, k]\}$. Let $G = (P, E(L) \cup S)$ be the graph obtained from the path L by adding to it all edges in S .

Finally, define $T^* = T_\theta^*$ to be an SPT over G rooted at rt .

We remark that the constructed tree T^* is similar to the original SLT construction of [5, 6].

Disregarding the time needed to compute the MST T , the graph G can be constructed within $O(n)$ time in the obvious way. Instead of taking an MST, we can use a constant-approximate MST – this will increase the lightness by only a constant factor. A constant-approximate MST can be built within $O(n)$ time in constant-dimensional Euclidean spaces [12]. Also, it is known that an SPT can be computed within linear time in planar graphs [30]. Observing that G is planar (but not necessarily plane), we conclude that T^* can be constructed within $O(n)$ time.

Next, we analyze the properties of the constructed tree T^* . Since T^* is an SPT over G rooted at rt , the root-stretch of (T^*, rt) is the same as that of (G, rt) . Also, the lightness of T^* is bounded above by that of G . It is therefore sufficient to establish the required bounds for G rather than for T^* .

For any pair of points v_i, v_j along L , with $i \leq j$, let $L(v_i, v_j)$ be the subpath of L between v_i and v_j . For each $i \in [k]$, denote by L_i the subpath of L between b_i and b_{i+1} , disregarding b_{i+1} , i.e., L_i contains all edges of $L(b_i, b_{i+1})$ except for the last one, and let P_i denote the set of points in P lying on that path. (The path L_k is between b_k and the last point v_n along L .) By definition, path L is obtained from the concatenation of the k paths $L(b_1, b_2), \dots, L(b_{k-1}, b_k), L(b_k, v_n)$, i.e., $L = L(b_1, b_2) \circ \dots \circ L(b_{k-1}, b_k) \circ L(b_k, v_n)$. Also, we have $P = \bigcup_{i \in [k]} P_i$.

Notice that the graph G can be partitioned into k edge-disjoint trees τ_1, \dots, τ_k , where τ_i is obtained from the union of edge $e_i = (rt, b_i)$ and path $L(b_i, b_{i+1})$. For each $i \in [k-1]$, let T_i be the spanning tree of the point set $P_i^+ = \{rt\} \cup P_i$ obtained from tree τ_i by removing the last edge along $L(b_i, b_{i+1})$, or equivalently, obtained as the union of edge e_i and path L_i . Also, let $T_k = \tau_k$. Observe that the union of the k trees T_1, \dots, T_k is a spanning subgraph of G , and, in particular, it spans the entire point set P .

In Appendix B we prove the following lemma. (The proof of this lemma follows similar lines as in [5, 6], and is provided in Appendix B for completeness.)

Lemma 2. 1. The root-stretch of each tree (T_i, rt) is at most $1 + 2\theta$, $i \in [k]$. (Thus the root-stretch of G is at most $1 + 2\theta$ as well.) 2. The lightness of G is $O(\frac{1}{\theta})$.

By substituting $\theta = c\sqrt{\epsilon}$ in Lemma 2, for a small constant $c < \frac{1}{10}$, we obtain the following proposition.

Proposition 3. The constructed tree $T^* = T_{c\sqrt{\epsilon}}^*$ is a spanning $(1 + O(\sqrt{\epsilon}), O(\sqrt{\frac{1}{\epsilon}}))$ -SLT for P rooted at rt . The runtime of this construction is $O(n)$.

We proceed to analyzing more properties of the SLT T^* from Proposition 3. Fix any index $i \in [k]$.

Let j and j' be the indices such that $P_i = \{v_j, \dots, v_{j'}\}$, with $j \leq j'$. Note that $b_i = v_j, b_{i+1} = v_{j'+1}$.

We next argue that the path distance (i.e., the distance in L) between all points in P_i is small with respect to $\omega(e_i) = \|rt, b_i\|$.

Lemma 4. $\omega(L_i) = d_L(b_i, v_{j'}) \leq 2c\sqrt{\epsilon} \cdot \omega(e_i)$.

Proof. Since $v_{j'}$ was not identified as a break-point, we have $d_L(b_i, v_{j'}) \leq c\sqrt{\epsilon} \cdot \|rt, v_{j'}\|$. By the triangle inequality,

$$\|rt, v_{j'}\| \leq \|rt, b_i\| + \|b_i, v_{j'}\| \leq \|rt, b_i\| + d_L(b_i, v_{j'}) \leq \|rt, b_i\| + c\sqrt{\epsilon} \cdot \|rt, v_{j'}\|,$$

and so

$$\|rt, v_{j'}\| \leq \frac{1}{1 - c\sqrt{\epsilon}} \cdot \|rt, b_i\| = \frac{1}{1 - c\sqrt{\epsilon}} \cdot \omega(e_i).$$

Note that for $\epsilon \leq \frac{1}{4}$ and $c < 1$, it holds that

$$\frac{c\sqrt{\epsilon}}{1 - c\sqrt{\epsilon}} \leq \frac{c\sqrt{\epsilon}}{1 - \sqrt{\epsilon}} \leq 2c\sqrt{\epsilon}.$$

Altogether,

$$\omega(L_i) = d_L(b_i, v_{j'}) \leq c\sqrt{\epsilon} \cdot \|rt, v_{j'}\| \leq \frac{c\sqrt{\epsilon}}{1 - c\sqrt{\epsilon}} \cdot \omega(e_i) \leq 2c\sqrt{\epsilon} \cdot \omega(e_i).$$

□

Lemma 4 implies the following corollary.

Corollary 5.

1. For any pair v, w of points in P_i , $\|v, w\| \leq 2c\sqrt{\epsilon} \cdot \omega(e_i)$.
2. For any point $v \in P_i$, $(1 - 2c\sqrt{\epsilon}) \cdot \omega(e_i) \leq \|rt, v\| \leq (1 + 2c\sqrt{\epsilon}) \cdot \omega(e_i)$.
3. $\omega(e_i) \leq MST(P_i^+) \leq \omega(T_i) = \omega(e_i) + \omega(L_i) \leq (1 + 2c\sqrt{\epsilon}) \cdot \omega(e_i) \leq 2 \cdot \omega(e_i)$.

Proof.

1. By the triangle inequality, for any pair v, w of points in P_i , $\|v, w\| \leq d_L(v, w) \leq \omega(L_i) \leq 2c\sqrt{\epsilon} \cdot \omega(e_i)$. (The last inequality follows from Lemma 4.)

2. By the previous assertion, $\|b_i, v\| \leq 2c\sqrt{\epsilon} \cdot \omega(e_i)$. Hence $\|rt, v\| \leq \|rt, b_i\| + \|b_i, v\| \leq (1 + 2c\sqrt{\epsilon}) \cdot \omega(e_i)$. Similarly, we have $\|rt, v\| \geq \|rt, b_i\| - \|b_i, v\| \geq (1 - 2c\sqrt{\epsilon}) \cdot \omega(e_i)$.

3. This assertion follows from the construction and Lemma 4.

□

Phase 2: Reducing the Root-Stretch using Steiner Points.

Recall that T^* is defined as an SPT over G rooted at rt . Our way for reducing the root-stretch of T^* will be to reduce the root-stretch of G .

We will show that by adding Steiner points, we can reduce the root-stretch of (G, rt) from $1 + O(\sqrt{\epsilon})$ to $1 + \epsilon$ while preserving lightness $O(\sqrt{\frac{1}{\epsilon}})$. The resulting Steiner graph \hat{G} will span all points of P and some additional Steiner points, and the ultimate Steiner SLT \hat{T} will be obtained as an SPT over \hat{G} rooted at rt .

Consider the partition of graph G into the k edge-disjoint trees τ_1, \dots, τ_k , where τ_i is obtained from the union of edge $e_i = (rt, b_i)$ and path $L(b_i, b_{i+1})$. Recall that $T_k = \tau_k$, and for each $i \in [k-1]$, T_i is the spanning tree of the point set $P_i^+ = \{rt\} \cup P_i$ obtained from τ_i by removing the last edge along $L(b_i, b_{i+1})$. We know (see Lemma 2) that the root-stretch of each tree (T_i, rt) is at most $1 + 2c\sqrt{\epsilon}$. We also know that the union of the k trees T_1, \dots, T_k spans the entire point set P . By reducing the root-stretch of each tree T_i to $1 + \epsilon$, the root-stretch of their union will be reduced to $1 + \epsilon$ as well.

In what follows we construct a Steiner tree \hat{T}_i rooted at rt with root-stretch $1 + \epsilon$ and weight $O(\omega(T_i))$, spanning P_i^+ and $O(|P_i| + \sqrt{\frac{1}{\epsilon}})$ additional Steiner points. By the third assertion of Corollary 5, $\omega(T_i) = \Theta(MST(P_i^+))$, thus we are actually looking for a Steiner $(1 + \epsilon, O(1))$ -SLT for P_i^+ rooted at rt .

The graph \hat{G} will be obtained from the union of the k Steiner trees $\hat{T}_1, \dots, \hat{T}_k$. As mentioned, the root-stretch of \hat{G} will be $1 + \epsilon$. The lightness will be fine too, as it is not much greater than that of G :

$$\omega(\hat{G}) = \sum_{i \in [k]} \omega(\hat{T}_i) = \sum_{i \in [k]} O(\omega(T_i)) = O\left(\sum_{i \in [k]} \omega(T_i)\right) = O(\omega(G)).$$

Fix an index $i \in [k]$. The description of the Steiner tree construction \hat{T}_i is divided into three parts.

Part 1: computing an isosceles triangle separating rt from P_i .

Let w be the point in P_i closest to rt , and define $r := \|rt, w\|$. By the second assertion of Corollary 5,

$$(1 - 2c\sqrt{\epsilon}) \cdot \omega(e_i) \leq r \leq \|rt, b_i\| = \omega(e_i). \quad (2)$$

Consider the isosceles triangle $\Delta = (rt, a, b)$ with apex rt , apex angle $\alpha := \sqrt{\epsilon}$ and side length r , such that the bisector of the apex angle of this triangle Δ is the straight line connecting rt and w . Observe that the endpoints a and b of the triangle base lie on the boundary of the circle centered at rt with radius r , and no point of P_i lies in the triangle's interior. (See Figure 1.(i) for an illustration.)

Consider the cone \mathcal{C}_α spanning an angle of α whose apex is at rt , defined by the two infinite rays coinciding with the sides (rt, a) and (rt, b) of triangle Δ .

Lemma 6. All points in P_i must belong to \mathcal{C}_α .

Proof. Observe that the length of the triangle base is equal to $2r \cdot \sin(\frac{\alpha}{2})$. It is easy to see that w is pretty far from the boundaries of \mathcal{C}_α – the distance between w and any point on the boundaries is equal to half the length of the triangle base, i.e., to $r \cdot \sin(\frac{\alpha}{2})$. (See Figure 1.(ii) for an illustration.) On the other hand, the first assertion of Corollary 5 implies that all points in P_i are pretty close to each other – the distance between w and all points of P_i is at most

$$2c\sqrt{\epsilon} \cdot \omega(e_i) \leq \frac{2c\sqrt{\epsilon}}{1 - 2c\sqrt{\epsilon}} \cdot r \leq 4c\sqrt{\epsilon} \cdot r < r \cdot \sin\left(\frac{\alpha}{2}\right).$$

(The first inequality follows from Equation (2), and the second inequality holds for $\epsilon < \frac{1}{4}$ and $c < \frac{1}{2}$; recalling that $\alpha = \sqrt{\epsilon}$, the third inequality holds for $c < \frac{1}{10}$.) \square

Lemma 6 implies the following corollary.

Corollary 7. For any point $v \in P_i$, the straight line connecting rt with v intersects the base of triangle $\triangle = (rt, a, b)$, at some point denoted \tilde{v} . (See Figure 2.(i) for an illustration.)

Part 2: distributing Steiner points on the triangle base and connecting them to P_i^+ to control root-stretch.

We distribute $\lceil \sqrt{\frac{1}{\epsilon}} \rceil$ evenly spaced Steiner points along the triangle base. Since the length of the triangle base is $2r \cdot \sin(\frac{\alpha}{2})$, the distance between any two consecutive Steiner points is at most $2r \cdot \sin(\frac{\alpha}{2}) \cdot \sqrt{\epsilon} = O(\epsilon) \cdot r$. For each point $v \in P_i$, consider the point \tilde{v} as defined in Corollary 7, and let v' be the Steiner point on the triangle base that is closest to \tilde{v} . Observe that $\|\tilde{v}, v'\| = O(\epsilon) \cdot r$ and $\|rt, v\| = \|rt, \tilde{v}\| + \|\tilde{v}, v\|$. By definition, $\|rt, v\| \geq r$. Consider the path $(rt, v', v) = (rt, v') \circ (v', v)$. By the triangle inequality,

$$\begin{aligned} \omega(rt, v', v) &= \|rt, v'\| + \|v', v\| \leq \|rt, \tilde{v}\| + \|\tilde{v}, v'\| + \|v', \tilde{v}\| + \|\tilde{v}, v\| \\ &= \|rt, v\| + 2 \cdot \|\tilde{v}, v'\| = \|rt, v\| + O(\epsilon) \cdot r \leq (1 + O(\epsilon)) \cdot \|rt, v\|, \end{aligned} \quad (3)$$

and so (rt, v', v) is a $(1 + O(\epsilon))$ -path between rt and v . This means that if we add the two edges (rt, v') and (v', v) for each point $v \in P_i$, the resulting stretch between rt and all points in P_i will be $1 + O(\epsilon)$.

Denote by V'_i the aforementioned set of $\lceil \sqrt{\frac{1}{\epsilon}} \rceil$ evenly spaced Steiner points lying on the triangle base. Let S_i be the star rooted at rt with points in V'_i serving as its leaves, i.e., $E(S_i) = \{(rt, v') \mid v' \in V'_i\}$. Also, let $E'_i = \{(v', v) \mid v \in P_i\}$ be the set of edges connecting each point $v \in P_i$ with its corresponding point v' as defined above. We can then define T'_i as the Steiner tree of P_i^+ obtained from the union of S_i and E'_i , i.e., $E(T'_i) = E(S_i) \cup E'_i$. (See Figure 2 for an illustration.) Equation (3) implies that the root-stretch of (T'_i, rt) is at most $1 + O(\epsilon)$. However, the lightness of tree T'_i is too large; in fact, it is much larger than that of the original tree T_i .

Part 3: reducing the lightness without hurting the root-stretch.

Reducing the lightness of tree T'_i (while preserving a low root-stretch) is carried out in two stages.

1. In the first stage we reduce the lightness contribution due to the edge set E'_i .

Lemma 8. For each edge $(v', v) \in E'_i$, $v \in P_i$, we have $\|v', v\| = O(\sqrt{\epsilon}) \cdot \omega(e_i)$.

Proof. Consider an arbitrary edge $(v', v) \in E'_i$, for $v \in P_i$. The first assertion of Corollary 5 implies that w is within distance $2c\sqrt{\epsilon} \cdot \omega(e_i)$ from all points of P_i , including v . Note also that w is within distance at most $\|w, a\| = \|w, b\| \approx r \cdot \sin(\frac{\alpha}{2}) = O(\sqrt{\epsilon}) \cdot \omega(e_i)$ from all points of V' , including v' . It follows that $\|v', v\| \leq \|v', w\| + \|w, v\| = O(\sqrt{\epsilon}) \cdot \omega(e_i)$. \square

We are going to keep just $O(\sqrt{\frac{1}{\epsilon}})$ edges of E'_i , throwing the others away. By Lemma 8, the total weight of the $O(\sqrt{\frac{1}{\epsilon}})$ edges that we keep will be bounded above by $O(\omega(e_i))$, yielding the desired lightness bound. To determine which edges should we keep, we use the following lemma.

Lemma 9. There is a point set $N_i \subset P_i$ of size $O(\sqrt{\frac{1}{\epsilon}})$ which “covers” the point set P_i in the following sense: Each point $v \in P_i$ has a point $p(v)$ from N_i satisfying $d_L(v, p(v)) \leq \epsilon \cdot \omega(e_i)$. (We say that N_i is an ϵ -net for P_i , and refer to the points of N_i as *net points*.) Moreover, we can compute this ϵ -net N_i in time $O(|P_i|)$.

Proof. By Lemma 4, the path distance (i.e., the distance in L_i) between all points in P_i is at most $2c\sqrt{\epsilon} \cdot \omega(e_i)$. This means that we can partition L_i into $\epsilon' = \lceil 2c\sqrt{\frac{1}{\epsilon}} \rceil$ vertex-disjoint subpaths $L_i^{(1)}, \dots, L_i^{(\epsilon')}$, each of weight at most $\epsilon \cdot \omega(e_i)$. Note that each subpath $L_i^{(j)}$ spans a subset of P_i , denoted $P_i^{(j)}$, and we have $P_i = \bigcup_{j \in [\epsilon']} P_i^{(j)}$. For each $j \in [\epsilon']$, we assign an arbitrary *net point* $p_i^{(j)} \in P_i^{(j)}$, and denote by $N_i = \{p_i^{(j)} \mid j \in [\epsilon']\}$ the set of all ϵ' net points. By definition, for each $j \in [\epsilon']$ and any point $v \in P_i^{(j)}$, the path distance (i.e., the distance in $L_i^{(j)}$) between v and the net point $p_i^{(j)}$ is at most $\epsilon \cdot \omega(e_i)$. Obviously, this ϵ -net N_i for P_i can be computed in time $O(|P_i|)$. \square

Next, we argue that it suffices to keep edges (v', v) of E'_i just for the ϵ' net points $v \in N_i$; we call these edges the *net edges* of E'_i . Indeed, for any $j \in [\epsilon']$, each point $v \in P_i^{(j)} \setminus N_i$ can be “glued” to its net point $p(v) = p_i^{(j)} \in N_i$ via the appropriate subpath $L_i^{(j)}$ of L_i , whose weight is at most $\epsilon \cdot \omega(e_i) = O(\epsilon) \cdot r$. Thus instead of taking a direct edge (v, v') , we can go from v to the net point $u := p_i^{(j)}$ via a short sub-path of $L_i^{(j)}$, namely $L(u, v)$, and then take a net edge (u, u') ; the following claim shows that this detour around u and u' will not be too costly.

Lemma 10. The path $(rt, u', u) \circ L(u, v)$ from rt to v is a $(1 + O(\epsilon))$ -path.

Proof. Recall that (rt, v', v) is a $(1 + O(\epsilon))$ -path between rt to v , and so

$$\omega(rt, v', v) = \|rt, v'\| + \|v', v\| \leq (1 + O(\epsilon)) \cdot \|rt, v\|. \quad (4)$$

Consider the points \tilde{u} and \tilde{v} , as defined in Corollary 7. It is easy to see that

$$\|\tilde{u}, \tilde{v}\| \leq \|u, v\| \leq d_L(u, v) = O(\epsilon) \cdot r.$$

Note also that both $\|u', \tilde{u}\|$ and $\|\tilde{v}, v'\|$ are bounded by $O(\epsilon) \cdot r$. By the triangle inequality, $\|u', v'\| \leq \|u', \tilde{u}\| + \|\tilde{u}, \tilde{v}\| + \|\tilde{v}, v'\| = O(\epsilon) \cdot r$. It follows that

$$\begin{aligned} \omega(rt, u', u) &= \|rt, u'\| + \|u', u\| \leq \|rt, v'\| + \|v', u'\| + \|u', v'\| + \|v', v\| + \|v, u\| \\ &= \omega(rt, v', v) + 2\|u', v'\| + \|u, v\| = \omega(rt, v', v) + O(\epsilon) \cdot r. \end{aligned} \quad (5)$$

Using Equations (4) and (5) and the fact that $\|rt, v\| \geq r$, we conclude that

$$\begin{aligned} \omega((rt, u', u) \circ L(u, v)) &= \omega(rt, u', u) + d_L(u, v) \leq \omega(rt, v', v) + O(\epsilon) \cdot r \\ &\leq (1 + O(\epsilon)) \cdot \|rt, v\|. \end{aligned}$$

□

Denote by $\hat{E}_i = \{(v', v) \mid v \in N_i\} \cup \bigcup_{j \in [\epsilon]} E(L_i^{(j)})$ the union of the edge set E'_i and the edges of all the subpaths $L_i^{(j)}$ of L_i ; this edge set will replace E'_i . By Lemma 10, the root-stretch will remain $1 + O(\epsilon)$. Moreover, the total weight of these edges is $O(\omega(e_i))$, so the lightness becomes constant.

2. In the second stage we reduce the lightness contribution due to the star S_i .

Note that the point set $\{rt\} \cup V'_i$ is exactly the core example from Section 2.1. We employ the “black-box” construction with which we are equipped to obtain a Steiner $(1 + \epsilon, O(1))$ -SLT T_i^{core} for the point set $\{rt\} \cup V'_i$ rooted at rt ; this tree T_i^{core} will replace the star S_i .

The loss is negligible: Since the root-stretch of T_i^{core} is $1 + \epsilon$, the distances between rt and all points of V'_i will grow by at most a factor of $1 + \epsilon$. As a result, the distances between rt and all points of P_i will grow by at most the same factor, implying that the root-stretch will remain $1 + O(\epsilon)$.

The gain, however, is significant: While the lightness of S_i is huge, the lightness of T_i^{core} is constant.

Let \hat{T}_i be the graph obtained from the union of the edges of tree T_i^{core} and the edge set \hat{E}_i , i.e., $E(\hat{T}_i) = E(T_i^{core}) \cup \hat{E}_i$. In light of the above, \hat{T}_i is a Steiner tree for P_i^+ with root-stretch $1 + O(\epsilon)$ and constant lightness. Also, we can reduce the root-stretch to $1 + \epsilon$ by scaling the lightness up by some constant.

The graph \hat{G} obtained from the union of the k Steiner trees $\hat{T}_1, \dots, \hat{T}_k$ has the desired root-stretch and lightness, and the ultimate Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLT \hat{T} is defined as an SPT over \hat{G} rooted at rt .

Theorem 11. *For any 2-dimensional Euclidean n -point set P , a designated point $rt \in P$ and a number $\epsilon < 1$, there exists a Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLT $\hat{T} = \hat{T}(P)$ for P rooted at rt .*

Remarks. (1) We may assume that $\epsilon \ll 1$. Indeed, for constant ϵ , we can simply use the spanning SLT constructions of [5, 6, 29] which guarantee root-stretch $1 + \epsilon$ and lightness

$$O(\frac{1}{\epsilon}) = O(\sqrt{\frac{1}{\epsilon}}) = O(1).$$

(2) As mentioned, the tradeoff between root-stretch $1 + \epsilon$ and lightness $O(\sqrt{\frac{1}{\epsilon}})$ guaranteed by this theorem is tight in the regime $\epsilon = \Omega(\frac{1}{n^2})$. Note also that the star over P rooted at rt (whose edge set is $\{(rt, v) \mid v \in P \setminus \{rt\}\}$) is a spanning SPT with lightness $O(n)$. Consequently, there is no reason to apply this theorem in the regime $\epsilon = o(\frac{1}{n^2})$, where the lightness bound $O(\sqrt{\frac{1}{\epsilon}})$ becomes super-linear in n .

3 Runtime Analysis

In this section we analyze the runtime of the construction as well as the number of Steiner points used.

In Section 3.1 we provide a naive $O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$ -time implementation of our construction. In Section 3.2 we demonstrate that a simple modification of our construction can be implemented in linear time.

3.1 A Naive Implementation

Number of Steiner Points. We argue that the SLT construction \hat{T} that is guaranteed by Theorem 11 contains at most $O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$ Steiner points.

Recall that \hat{T} is defined as an SPT over the graph \hat{G} rooted at rt , and that \hat{G} is obtained from the union of the k trees $\hat{T}_1, \dots, \hat{T}_k$, with $k \leq n$.

By construction, each \hat{T}_i is a Steiner tree for P_i^+ . Moreover, all Steiner points of \hat{T}_i belong to the tree T_i^{core} , and all points in V'_i are leaves of T_i^{core} . Hence, Observation 1 implies that the number of Steiner points in \hat{T}_i is bounded by $O(|V'_i|) = O(\sqrt{\frac{1}{\epsilon}})$. Summing over all $k \leq n$ trees $\hat{T}_1, \dots, \hat{T}_k$, we conclude that the number of Steiner points in the ultimate SLT \hat{T} is at most $k \cdot O\left(\sqrt{\frac{1}{\epsilon}}\right) = O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$.

Runtime. In Phase 1 of the construction, we built a spanning SLT $T_{c\sqrt{\epsilon}}^*$. By Proposition 3, the time needed to build this tree is $O(n)$.

In Phase 2 of the construction, the main goal is to transform each spanning tree T_i of P_i^+ into a Steiner tree \hat{T}_i , for $i \in [k]$. We divided the description of this phase into three parts.

In Part 1 we computed an isosceles triangle separating rt from P_i , which boils down to finding a point w in P_i that is closest to rt , and can be done in $O(|P_i|)$ time. We also computed the intersection point of the straight line connecting rt and v and the triangle base, for each $v \in P_i$, which takes $O(|P_i|)$ time too.

In Part 2 we distributed $|V'_i| = \lceil \sqrt{\frac{1}{\epsilon}} \rceil$ Steiner points on the triangle base, and connected them to P_i^+ to form a Steiner tree T'_i for P_i^+ ; both these tasks can be easily

carried out within $O(|V'_i| + |P_i|)$ time.

In Part 3 we transformed the “heavy” tree T'_i constructed in Part 2 into the much “lighter” tree \hat{T}_i . This was carried out in two stages.

1. The bottleneck of the first stage was to compute an ϵ -net N_i for P_i . By Lemma 9, this takes $O(|P_i|)$ time. Equipped with this ϵ -net N_i , computing the corresponding net edges (i.e., the edges between the net points of N_i and the corresponding points from V'_i) can be done in another $O(|N_i|) = O(|P_i|)$ time.
2. In the second stage we computed a tree T_i^{core} for the point set $\{rt\} \cup V'_i$, using the construction described in Section 2.1 for the core example. By Observation 1, this can be done in $O(|V'_i|)$ time.

Thus for each $i \in [k]$, the runtime is bounded by $O(|V'_i| + |P_i|) = O(\sqrt{\frac{1}{\epsilon}} + |P_i|)$. Summing over all k indices, with $k \leq n$, the overall time needed to compute the Steiner trees $\hat{T}_1, \dots, \hat{T}_k$ is no greater than

$$\begin{aligned} \sum_{i \in [k]} O\left(\sqrt{\frac{1}{\epsilon}} + |P_i|\right) &= O\left(k \cdot \sqrt{\frac{1}{\epsilon}}\right) + O\left(\sum_{i \in [k]} |P_i|\right) \\ &= O\left(k \cdot \sqrt{\frac{1}{\epsilon}}\right) + O(n) = O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right). \end{aligned} \quad (6)$$

The time needed to compute the graph \hat{G} , obtained as the union of these trees, is proportional to the total number $O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$ of vertices. Finally, it is easy to see that graph \hat{G} is planar. As mentioned, an SPT can be computed in $O(n)$ time in planar graphs [30], which implies that the ultimate Steiner SLT \hat{T} can be extracted from \hat{G} in linear time.

Summarizing, the total runtime of our SLT construction \hat{T} is $O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$.

3.2 A Linear-Time Implementation

In Section 3.1 we showed that the tree \hat{T}_i can be constructed in time $O(|V'_i| + |P_i|) = O(\sqrt{\frac{1}{\epsilon}} + |P_i|)$, for each index $i \in [k]$. We would like to reduce the runtime of this construction to $O(|P_i|)$. (As the number of Steiner points is a lower bound on the runtime, we should keep at most $O(|P_i|)$ Steiner points in \hat{T}_i .) If we manage to do this, the runtime of the SLT construction \hat{T} (and the number of Steiner points used) will be reduced from $O\left(n \cdot \sqrt{\frac{1}{\epsilon}}\right)$ to $\sum_{i \in [k]} O(|P_i|) = O\left(\sum_{i \in [k]} |P_i|\right) = O(n)$ (cf. Equation (6)).

Number of Steiner Points. Fix an arbitrary index $i \in [k]$. As mentioned, all Steiner points of \hat{T}_i belong to the tree T_i^{core} , and moreover, all points in V'_i are leaves of T_i^{core} .

We would like to keep in T_i^{core} just $O(|P_i|)$ points. To this end, we will keep a Steiner point of T_i^{core} only if it lies on a path from rt to some point of P_i in \hat{T}_i . Such a point is called *useful*, and the non-useful Steiner points (that we will not keep) are called *redundant*.

By definition, an internal vertex in T_i^{core} is useful if and only if one of its descendant

leaves is useful. Thus distinguishing between useful and redundant Steiner points boils down to determining the useful points in V'_i . To this end we use the the following observation, which is immediate from the construction.

Observation 12. Each point v' of V'_i is useful if and only if it is incident on a net edge (v', v) , where $v \in N_i$ is a net point. In particular, the number of useful leaves in T_i^{core} is bounded by $|N_i| \leq |P_i|$.

Note that we can remove the redundant Steiner points of tree T_i^{core} via a straightforward bottom-up traversal of the tree. The resulting tree, however, may still contain too many Steiner points.

To reduce the size of the tree to $O(|P_i|)$, we contract all single-child paths. As a result, the number of internal vertices will be smaller than the number of leaves, which is, in turn, bounded by $|N_i| \leq |P_i|$. In other words, the size of the resulting *pruned* tree $Prune(T_i^{core})$ will be at most $2|P_i| - 1$, as required.

By plugging the resulting tree $Prune(T_i^{core})$ instead of T_i^{core} , we obtain a tree $Prune(\hat{T}_i)$ that spans all points in P_i^+ and contains just $O(|P_i|)$ Steiner points. Observe that this pruning procedure may only decrease the root-stretch and the lightness of the tree.

We apply this pruning procedure on each of the k trees $\hat{T}_1, \hat{T}_2, \dots, \hat{T}_k$. By taking the union of the resulting pruned trees $Prune(\hat{T}_1), Prune(\hat{T}_2), \dots, Prune(\hat{T}_k)$, we get a pruned graph $Prune(\hat{G})$ with only $\sum_{i=1}^k O(|P_i|) = O(n)$ Steiner points. Therefore, the pruned SLT $Prune(\hat{T})$ that we get in this way (defined as an SPT over the pruned graph $Prune(\hat{G})$ rooted at rt) will have $O(n)$ Steiner points as well.

Runtime. To achieve a linear-time implementation, reducing the size of each tree \hat{T}_i to $O(|P_i|)$ is not enough – we need to be able to construct the corresponding pruned tree $Prune(\hat{T}_i)$ within time $O(|P_i|)$.

Consequently, we cannot afford to construct the original tree \hat{T}_i explicitly. (Note that even distributing the $|V'_i| = \lceil \sqrt{\frac{1}{\epsilon}} \rceil$ Steiner points on the triangle base in Part 2 of the construction may be too costly.)

We next show how to build the pruned tree $Prune(\hat{T}_i)$ *directly*, i.e., without using the original tree \hat{T}_i that may contain too many Steiner points. It suffices to show how to construct the tree $Prune(T_i^{core})$ directly, within $O(|P_i|)$ time. (Having constructed the tree $Prune(T_i^{core})$, the tree $Prune(\hat{T}_i)$ is obtained from it by adding the net edges, which by Lemma 9 requires $O(|P_i|)$ time.)

Denote by U_i the set of vertices in the pruned tree $Prune(T_i^{core})$. Also, denote by $Leaves(U_i)$ the set of leaves in $Prune(T_i^{core})$, and note that $Leaves(U_i)$ also designates the set of useful Steiner leaves of the original tree T_i^{core} . By Observation 12 and Lemma 9, the vertex set $Leaves(U_i)$ can be computed within time $O(|P_i|)$. In order to compute the remaining vertices of U_i , we use the following observation.

Observation 13. Any vertex of U_i is a *least common ancestor* in the tree T_i^{core} of a pair u, v of consecutive vertices in $Leaves(U_i)$.

Note also that the pruned tree $Prune(T_i^{core})$ preserves the hierarchical structure of

the original tree T_i^{core} : The parent of any vertex v in $Prune(T_i^{core})$ is the nearest ancestor of v in T_i^{core} that belongs to U_i .

The main hurdle on the way to constructing tree $Prune(T_i^{core})$ in time $O(|P_i|)$ is to be able to compute its vertex set U_i within this time bound. We remark that this is not a geometric issue; even though this hurdle lies outside the scope of the current paper, below we describe how it can be resolved.

Note that, disregarding the locations of the points in the plane, the structure of tree T_i^{core} is independent of the point set P_i : (1) It contains precisely $|V_i'| = \lceil \sqrt{\frac{1}{\epsilon}} \rceil$ leaves. (2) By contracting the leaves of this tree with their parents (each of these parent has a single child), we get a complete binary tree.

Although we cannot build the tree T_i^{core} explicitly for each $i, i \in [k]$, we can construct a single skeleton T_{bin} of a complete binary tree with $\lceil \sqrt{\frac{1}{\epsilon}} \rceil$ leaves, which will be used for answering least common ancestor queries efficiently: It is well-known that any complete binary tree on n vertices can be preprocessed in $O(n)$ time to answer least common ancestor queries in constant time. (See, e.g., Chapter 2 in [39].) By the second remark following Theorem 11, we may assume that the number $\lceil \sqrt{\frac{1}{\epsilon}} \rceil$ of leaves in T_{bin} is at most n . Constructing this tree T_{bin} requires $O(\lceil \sqrt{\frac{1}{\epsilon}} \rceil) = O(n)$ time, but we only do it once.

To compute the vertex set U_i , we start by identifying the leaves of T_{bin} that correspond to the vertices of $Leaves(U_i)$, and inserting them into a linked list A_0 in the same order that they appear in the tree. This task can be carried out in $O(|P_i|)$ time in the obvious way.

For any pair u, v of consecutive leaves in A_0 , we can find their least common ancestor z in T_{bin} within constant time. If z is at level ℓ in tree T_{bin} , then we insert it into list A_ℓ . (We say that a vertex is at level ℓ of the tree, for any $\ell \geq 0$, if its minimum unweighted distance from any leaf is equal to ℓ .)

By Observation 13, at the end of this process, the union of the non-empty lists A_ℓ , $\ell \geq 0$, gives U_i . Moreover, this process can be easily implemented within time $O(|U_i|) = O(|P_i|)$.

Next, we need to compute the parent of each vertex $v \in U_i$ in the tree $Prune(T_i^{core})$. This can be carried out via a simple bottom-up process, in which we do not traverse any of the trees directly (as this would require too much time), but rather use the non-empty lists A_ℓ , $\ell \geq 0$, to access just the vertices of U_i . If performed carefully, this process will increase the overall runtime by at most a constant factor.

Summarizing, the pruned tree $Prune(T_i^{core})$ can be constructed within time $O(|P_i|)$. Consequently, the runtime of the ultimate SLT construction $Prune(\hat{T})$ that we get in this way will be reduced to $O(n)$.

The main result of this paper is summarized in the following theorem.

Theorem 14. *For any 2-dimensional Euclidean n -point set P , a designated point $rt \in P$ and a number $\epsilon < 1$, there exists a Steiner $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLT $Prune(\hat{T}) = Prune(\hat{T}(P))$*

for P rooted at rt , having $O(n)$ Steiner points. Moreover, the runtime of this construction is $O(n)$.

4 Conclusions and Open Questions

In this paper we showed that Steiner points help significantly in the context of Euclidean SLTs.

We anticipate that the ideas used here will be applicable to other geometric problems that involve stretch or weight in Steiner trees.

There are various natural directions for future research, we mention below just three of them.

The first question is whether one can obtain Euclidean $(1 + \epsilon, O(\sqrt{\frac{1}{\epsilon}}))$ -SLTs in Euclidean spaces of any dimension $d, d \geq 2$. In the proceedings version of this paper we outlined the main ideas needed for extending the construction of Section 2 to higher-dimensional Euclidean spaces $d \geq 2$. However, the lightness bound of the resulting construction will be $O(\sqrt{\frac{1}{\epsilon}}) \cdot 2^d$ rather than $O(\sqrt{\frac{1}{\epsilon}})$, i.e., there is an exponential dependence on the dimension d in the lightness bound. We conjecture that the 2-dimensional case is the hardest one, and that the correct lightness in any dimension is $O(\sqrt{\frac{1}{\epsilon}})$.

Our results for 2-dimensional Euclidean spaces imply that, for any $\epsilon > 0$, one can get a Steiner SLT with root-stretch $1 + \epsilon$ and lightness at most $\zeta \cdot \sqrt{\frac{1}{\epsilon}}$, for an appropriate constant ζ . An intriguing question in this context is to determine the optimal value for the leading constant ζ . In fact, this question is also open in the more basic setting of Euclidean spanning SLTs (with no Steiner points), with the lightness bound there being $\zeta \cdot \frac{1}{\epsilon}$ rather than $\zeta \cdot \sqrt{\frac{1}{\epsilon}}$ (for a possibly different leading constant).

Finally, we believe that Steiner points should lead to a quadratic improvement in the context of *light Euclidean spanners*. More specifically, it is known for many years that a *spanning* $(1 + \epsilon)$ -spanner (with no Steiner points) of lightness $O((\frac{1}{\epsilon})^{2d})$ exists in any d -dimensional Euclidean space, and there is also a similar lower bound. (See Chapter 15 in [39] for details.) We anticipate that, by combining the ideas presented in this paper with known constructions of light spanning spanners, one could get a *Steiner* $(1 + \epsilon)$ -spanner with lightness $O((\frac{1}{\epsilon})^d)$.

Acknowledgments

The author is grateful to Michael Elkin for helpful discussions.

References

- [1] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. *Contemporary Mathematics*, 223:1–56, 1999.

- [2] N. Alon, R. M. Karp, D. Peleg, and D. B. West. A graph-theoretic game and its application to the k -server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.
- [3] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger. Prim-Dijkstra tradeoffs for improved performance-driven routing tree design. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 14(7):890–896, 1995.
- [4] B. Aronov, M. de Berg, O. Cheong, J. Gudmundsson, H. J. Haverkort, M. H. M. Smid, and A. Vigneron. Sparse geometric graphs with small dilation. *Comput. Geom.*, 40(3):207–219, 2008.
- [5] B. Awerbuch, A. Baratz, and D. Peleg. Cost-sensitive analysis of communication protocols. In *Proc. of 9th PODC*, pages 177–187, 1990.
- [6] B. Awerbuch, A. Baratz, and D. Peleg. Efficient broadcast and light-weight spanners. *Manuscript*, 1991.
- [7] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proc. of 37th FOCS*, pages 184–193, 1996.
- [8] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proc. of 30th STOC*, pages 161–168, 1998.
- [9] Y. Ben-Shimol, A. Dvir, and M. Segal. SPLAST: a novel approach for multicasting in mobile wireless ad hoc networks. In *Proc. of 15th PIMRC*, pages 1011–1015, 2004.
- [10] R. Braynard, D. Kotic, A. Rodriguez, J. Chase, and A. Vahdat. Opus: an overlay peer utility service. In *Proc. of 5th OPENARCH*, 2002.
- [11] P. Carmi and L. Chaitman-Yerushalmi. Unexplored steiner ratios in geometric networks. In *Proc. of 18th COCOON*, pages 275–286, 2012.
- [12] T. M. Chan. Well-separated pair decomposition in linear time? *Inf. Process. Lett.*, 107(5):138–141, 2008.
- [13] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *J. Algorithms*, 33(1):73–91, 1999.
- [14] F. Chung and R. Graham. A new bound for euclidean steiner minimal trees. *Annals of the New York Academy of Sciences*, 440(1):328–346, 1985.
- [15] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Performance-driven global routing for cell based ics. In *Proc. of 9th ICCD*, pages 170–173, 1991.
- [16] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Provably good algorithms for performance-driven global routing. In *Proc. of 5th ISCAS*, pages 2240–2243, 1992.
- [17] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. Provably good performance-driven global routing. *IEEE Trans. on CAD of Integrated Circuits and Sys.*, 11(6):739–752, 1992.

- [18] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer. Network correlated data gathering with explicit communication: NP-completeness and algorithms. *IEEE/ACM Trans. Netw.*, 14(1):41–54, 2006.
- [19] Y. Dinitz, M. Elkin, and S. Solomon. Low-light trees, and tight lower bounds for Euclidean spanners. *Discrete & Computational Geometry*, 43(4):736–783, 2010.
- [20] M. Elkin and S. Solomon. Narrow-shallow-low-light trees with and without Steiner points. *SIAM J. Discrete Math.*, 25(1):181–210, 2011.
- [21] M. Elkin and S. Solomon. Steiner shallow-light trees are exponentially lighter than spanning ones. In *Proc. of 52nd FOCS*, pages 373–382, 2011. The journal version of this paper is currently under review.
- [22] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. of 35th STOC*, pages 448–455, 2003.
- [23] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM J. Applied Math.*, 16(1):1–29, 1968.
- [24] A. Goel and K. Munagala. Balancing Steiner trees and shortest path trees online. In *Proc. of 11th SODA*, pages 562–563, 2000.
- [25] A. Gupta. Steiner points in tree metrics don’t (really) help. In *Proc. of 12th SODA*, pages 220–227, 2001.
- [26] M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximating buy-at-bulk and shallow-light k -Steiner trees. *Algorithmica*, 53(1):89–103, 2009.
- [27] A. O. Ivanov and A. A. Tuzhilin. The Steiner Ratio Gilbert-Pollak Conjecture Is Still Open - Clarification Statement. *Algorithmica*, 62(1-2), 630-632, 2012.
- [28] X. Jia, D.-Z. Du, X.-D. Hu, M.-K. Lee, and J. Gu. Optimization of wavelength assignment for QoS multicast in WDM networks. *IEEE Transactions on Communications*, 49(2):341–350, 2001.
- [29] S. Khuller, B. Raghavachari, and N. E. Young. Balancing minimum spanning and shortest path trees. In *Proc. of 4th SODA*, pages 243–250, 1993.
- [30] P. N. Klein, S. Rao, M. Rauch, and S. Subramanian. Faster shortest-path algorithms for planar graphs. In *Proc. of 26th STOC*, pages 27–37, 1994.
- [31] V. P. Kompella, J. Pasquale, and G. C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Trans. Netw.*, 1(3):286–292, 1993.
- [32] G. Konjevod, R. Ravi, and F. S. Salman. On approximating planar metrics by tree metrics. *Inf. Process. Lett.*, 80(4):213–219, 2001.
- [33] G. Kortsarz and D. Peleg. Approximating the weight of shallow Steiner trees. *Discrete Applied Mathematics*, 93(2-3):265–285, 1999.

- [34] D. Kostic and A. Vahdat. Latency versus cost optimizations in hierarchical overlay networks. *Technical report, Duke University*, (CS-2001-04), 2002.
- [35] W. Liang and Y. Liu. Online data gathering for maximizing network lifetime in sensor networks. *IEEE Trans. Mob. Comput.*, 6(1):2–11, 2007.
- [36] H. Luo, J. Luo, Y. Liu, and S. K. Das. Adaptive data fusion for energy efficient routing in wireless sensor networks. *IEEE Trans. Computers*, 55(10):1286–1299, 2006.
- [37] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III. Bicriteria network design problems. *J. Algorithms*, 28(1):142–171, 1998.
- [38] J. Naor and B. Schieber. Improved approximations for shallow-light spanning trees. In *Proc. of 38th FOCS*, pages 536–541, 1997.
- [39] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [40] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, Philadelphia, PA, 2000.
- [41] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Buy-at-bulk network design: Approximating the single-sink edge installation problem. In *Proc. of 8th SODA*, pages 619–628, 1997.
- [42] A. Shaikh and K. G. Shin. Destination-driven routing for low-cost multicast. *IEEE Journal on Selected Areas in Communications*, 15(3):373–381, 1997.
- [43] H. Shpungin and M. Segal. Near optimal multicriteria spanner constructions in wireless ad-hoc networks. *IEEE/ACM Transactions on Networking*, 18(6):1963–1976, 2010.
- [44] J. Vogel, J. Widmer, D. Farin, M. Mauve, and W. Effelsberg. Priority-based distribution trees for application-level multicast. In *Proc. of 2nd NETGAMES*, pages 148–157, 2003.
- [45] P. von Rickenbach and R. Wattenhofer. Gathering correlated data in sensor networks. In *Proc. of DIALM-POMC*, pages 60–66, 2004.
- [46] B. Y. Wu, K.-M. Chao, and C. Y. Tang. Light graphs with small routing cost. *Networks*, 39(3):130–138, 2002.

A A Lower Bound for Euclidean Steiner SLTs from [21]

In this appendix we describe a lower bound from the FOCS'11 paper [21]. The details of this lower bound were deferred to the journal version of [21], which is currently under review. These details are provided here for the sake of completeness.

Let C_n denote a set of n points that are uniformly spaced on the boundary of the unit circle C centered at the origin, and define $\tilde{C}_{n+1} = C_n \cup \{(0,0)\}$. We will show that any Euclidean Steiner tree (T, rt) for \tilde{C}_{n+1} rooted at $rt = (0,0)$ with root-stretch at most $1 + \epsilon$ must have lightness at least $\Omega(\sqrt{\frac{1}{\epsilon}})$, for $\epsilon = \Omega(\frac{1}{n^2})$.

Partition the circle C into $t = \Theta(\sqrt{\frac{1}{\epsilon}})$ arcs A_1, \dots, A_t of angle $\frac{2\pi}{t} = \Theta(\sqrt{\epsilon})$ each. Since $\epsilon = \Omega(\frac{1}{n^2})$, we may assume that each arc A_i contains at least $\Omega(1)$ points of C_n . Consider two arbitrary points p_i and p_j that reside at the middle of two distinct arcs A_i and A_j , respectively. They are at circular distance at least $\Omega(\sqrt{\epsilon})$ from each other. As the root-stretch is at most $1 + \epsilon$, we know that the stretches in T between rt and p_i and between rt and p_j are at most $1 + \epsilon$.

Lemma 15. The paths in T from rt to p_i and from rt to p_j cannot intersect within the annulus \mathcal{A} with inner radius $\frac{1}{2}$ and outer radius 1 centered at the origin.

Proof. Let q_i (respectively, q_j) be the point lying in the middle of the segment connecting rt with p_i (respectively, p_j); note that q_i and q_j lie on the boundary of the annulus \mathcal{A} . Let q be the point lying in the middle of the arc connecting q_i and q_j on the boundary of \mathcal{A} . Observe that the circular distance between q_i and q_j is $\Omega(\sqrt{\epsilon})$. By symmetry considerations, if the tree paths between rt and p_i and between rt and p_j intersect within the annulus \mathcal{A} , the maximum root-stretch is minimized if the intersection point is q . However, in the latter case, a straightforward calculation shows that the root-stretch is $\|rt, q\| + \|q, p_i\| = \frac{1}{2} + \|q, p_i\| \geq 1 + \Omega(\epsilon)$. By approximately setting the constant hidden by the O -notation in $t = O(\sqrt{\frac{1}{\epsilon}})$ we can guarantee here root-stretch greater than $1 + \epsilon$. \square

It follows that each point p_i at the middle of arc A_i contributes $\frac{1}{2}$ fresh units to the weight of T , for each $i = [t]$. Since there are $t = \Theta(\sqrt{\frac{1}{\epsilon}})$ such arcs, it follows that the weight of T is at least $\frac{1}{2} \cdot t = \Omega(\sqrt{\frac{1}{\epsilon}})$. Note also that the weight of the MST for \tilde{C}_{n+1} is $O(1)$, which completes the argument.

Remark. The above lower bound for \tilde{C}_{n+1} holds for the specific choice of root vertex $rt = (0,0)$. However, this argument can be easily strengthened to hold for C_n and *any* choice of root vertex $rt \in C_n$.

B Proof of Lemma 2

In this appendix we provide the proof of Lemma 2 from Section 2.2. As mentioned, this proof follows similar lines as those in the works of [5, 6], and is provided here for the sake

of completeness.

The first assertion of Lemma 2 follows from the next claim.

Claim 16. The root-stretch of each tree (T_i, rt) is at most $1 + 2\theta$, $i \in [k]$.

Proof. As T_i has a direct edge e_i between rt and b_i , $d_{T_i}(rt, b_i) = \|rt, b_i\|$. Consider any point $v \in P_i \setminus \{b_i\}$, and let $\Pi_v := e_i \circ L(b_i, v)$ be the path in T_i between rt and v obtained by concatenating edge $e_i = (rt, b_i)$ with the subpath $L(b_i, v)$ of L between b_i and v . Since v was not identified as a break-point, necessarily

$$d_L(b_i, v) \leq \theta \cdot \|rt, v\|, \quad (7)$$

and so

$$d_{T_i}(rt, v) = \omega(\Pi_v) = \|rt, b_i\| + d_L(b_i, v) \leq \|rt, b_i\| + \theta \cdot \|rt, v\|. \quad (8)$$

By the triangle inequality and Equation (7),

$$\|rt, b_i\| \leq \|rt, v\| + \|b_i, v\| \leq \|rt, v\| + d_L(b_i, v) \leq (1 + \theta) \cdot \|rt, v\|. \quad (9)$$

Plugging Equation (9) in Equation (8), we obtain

$$d_{T_i}(rt, v) \leq \|rt, b_i\| + \theta \cdot \|rt, v\| \leq (1 + \theta) \cdot \|rt, v\| + \theta \cdot \|rt, v\| = (1 + 2\theta) \cdot \|rt, v\|.$$

□

Recall that $\omega(L) \leq 2 \cdot \omega(MST(P))$. The next claim shows that the lightness of G is at most $O(\frac{1}{\theta})$, thus proving the second assertion of Lemma 2.

Claim 17. $\omega(G) \leq (1 + \frac{1}{\theta}) \cdot \omega(L)$.

Proof. By construction, $\omega(G) = \omega(L) + \sum_{i=1}^{k-1} \omega(e_{i+1})$. The choice of break-points implies that $\omega(e_{i+1}) = \|rt, b_{i+1}\| < \frac{1}{\theta} \cdot d_L(b_i, b_{i+1})$, for each index $i \in [k-1]$. Since $L = L(b_1, b_2) \circ \dots \circ L(b_{k-1}, b_k) \circ L(b_k, v_n)$, it holds that

$$\sum_{i=1}^{k-1} d_L(b_i, b_{i+1}) \leq \sum_{i=1}^{n-1} d_L(v_i, v_{i+1}) = \sum_{i=1}^{n-1} \|v_i, v_{i+1}\| = \omega(L).$$

Therefore,

$$\sum_{i=1}^{k-1} \omega(e_{i+1}) = \sum_{i=1}^{k-1} \|rt, b_{i+1}\| < \frac{1}{\theta} \cdot \sum_{i=1}^{k-1} d_L(b_i, b_{i+1}) \leq \frac{1}{\theta} \cdot \omega(L).$$

It follows that

$$\omega(G) = \omega(L) + \sum_{i=1}^{k-1} \omega(e_{i+1}) \leq \left(1 + \frac{1}{\theta}\right) \cdot \omega(L).$$

□