Minimum Weight Euclidean $(1+\varepsilon)$-Spanners*

Csaba D. Tóth[1] [0000−0002−8769−3190]

[1] California State University Northridge, Los Angeles, CA, USA
[2] Tufts University, Medford, MA, USA
csaba.toth@csun.edu

Abstract. Given a set S of n points in the plane and a parameter $\varepsilon > 0$, a Euclidean $(1+\varepsilon)$-spanner is a geometric graph $G = (S, E)$ that contains a path of weight at most $(1+\varepsilon)\|pq\|_2$ for all $p, q \in S$. We show that the minimum weight of a Euclidean $(1+\varepsilon)$-spanner for n points in the unit square $[0,1]^2$ is $O(\varepsilon^{-3/2}\sqrt{n})$, and this bound is the best possible. The upper bound is based on a new spanner algorithm that sparsifies Yao-graphs. It improves upon the baseline $O(\varepsilon^{-2}\sqrt{n})$, obtained by combining a tight bound for the weight of an MST and a tight bound for the lightness of Euclidean $(1+\varepsilon)$-spanners, which is the ratio of the spanner weight to the weight of the MST. The result generalizes to d-space for all $d \in \mathbb{N}$: The minimum weight of a Euclidean $(1+\varepsilon)$-spanner for n points in the unit cube $[0,1]^d$ is $O_d(\varepsilon^{(1-d^2)/d} n^{(d-1)/d})$, and this bound is the best possible. For the $n \times n$ section of the integer lattice, we show that the minimum weight of a Euclidean $(1+\varepsilon)$-spanner is between $\Omega(\varepsilon^{-3/4} n^2)$ and $O(\varepsilon^{-1}\log(\varepsilon^{-1})\, n^2)$. These bounds become $\Omega(\varepsilon^{-3/4}\sqrt{n})$ and $O(\varepsilon^{-1}\log(\varepsilon^{-1})\sqrt{n})$ when scaled to a grid of n points in $[0,1]^2$.

Keywords: Geometric spanner · Yao-graph · Farey sequences.

1 Introduction

For a set S of n points in a metric space, a graph $G = (S, E)$ is a *t-spanner* if G contains, between any two points $p, q \in S$, a pq-path of weight at most $t \cdot \|pq\|$, where $t \geq 1$ is the *stretch factor* of the spanner. In other words, a t-spanner approximates the true distances between the $\binom{n}{2}$ pairs of points up to a factor t distortion. Several optimization criteria have been developed for t-spanners for a given parameter $t \geq 1$. Natural parameters are the *size* (number of edges), the *weight* (sum of edge weights), the *maximum degree*, and the *hop-diameter*. Specifically, the *sparsity* of a spanner is the ratio $|E|/|S|$ between the number of edges and vertices; and the *lightness* is the ratio between the weight of a spanner and the weight of an MST on S.

In the geometric setting, S is a set of n points in Euclidean d-space in constant dimension $d \in \mathbb{N}$. For every $\varepsilon > 0$, there exist $(1+\varepsilon)$-spanners with $O_d(\varepsilon^{1-d})$ sparsity and $O_d(\varepsilon^{-d})$ lightness, and both bounds are the best possible [23]. In

* Research partially supported by NSF grant DMS-1800734.

particular, the Θ-graphs, Yao-graphs [31], gap-greedy and path-greedy spanners provide $(1+\varepsilon)$-spanners of sparsity $O_d(\varepsilon^{1-d})$. For lightness, Das et al. [9,10,28] were the first to construct $(1+\varepsilon)$-spanners of lightness $\varepsilon^{-O(d)}$. Gottlieb [18] generalized this result to metric spaces with doubling dimension d; see also [6,15]. Recently, Le and Solomon [23] showed that the greedy $(1+\varepsilon)$-spanner in $\mathbb{R}^d$ has lightness $O(\varepsilon^{-d})$; and so it simultaneously achieves the best possible bounds for both lightness and sparsity. The greedy $(1+\varepsilon)$-spanner algorithm [4] generalizes Kruskal's algorithm: It sorts the $\binom{n}{2}$ edges of K_n by nondecreasing weight, and incrementally constructs a spanner H: it adds an edge uv if H does not contain an uv-path of weight at most $(1+\varepsilon)\|uv\|$.

Lightness versus Minimum Weight. Lightness is a convenient optimization parameter, as it is invariant under scaling. It also provides an approximation ratio for the minimum weight $(1+\varepsilon)$-spanner, as the weight of a Euclidean MST (for short, EMST) is a trivial lower bound on the spanner weight. However, minimizing the lightness is not equivalent to minimizing the spanner weight for a given input instance, as the EMST is highly sensitive to the distribution of the points in S. Given that worst-case tight bounds are now available for the lightness, it is time to revisit the problem of approximating the *minimum weight* of a Euclidean $(1+\varepsilon)$-spanner, without using the EMST as an intermediary.

Euclidean Minimum Spanning Trees. For n points in the unit cube $[0,1]^d$, the weight of the EMST is $O_d(n^{1-1/d})$, and this bound is also best possible [14,33]. In particular, a suitably scaled section of the integer lattice attains these bounds up to constant factors. Supowit et al. [34] proved similar bounds for the minimum weight of other popular graphs, such as spanning cycles and perfect matchings on n points in the unit cube $[0,1]^d$.

Extremal Configurations for Euclidean $(1+\varepsilon)$-Spanners. The tight $O_d(\varepsilon^{-d})$ bound on lightness [23] implies that for every set of n points in $[0,1]^d$, there is a Euclidean $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-d}n^{1-1/d})$. However, the combination of two tight bounds need not be tight; and it is unclear which n-point configurations require the heaviest $(1+\varepsilon)$-spanners. We show that this bound can be improved to $O(\varepsilon^{-3/2}\sqrt{n})$ in the plane. Furthermore, the extremal point configurations are not an integer grid, but an asymmetric grid.

Contributions. We obtain a tight upper bound on the minimum weight of a Euclidean $(1+\varepsilon)$-spanner for n points in $[0,1]^d$.

Theorem 1. *For constant $d \geq 2$, every set of n points in the unit cube $[0,1]^d$ admits a Euclidean $(1+\varepsilon)$-spanner of weight $O_d(\varepsilon^{(1-d^2)/d}n^{(d-1)/d})$, and this bound is the best possible.*

The upper bound is established by a new spanner algorithm, SparseYao, that sparsifies the classical Yao-graph using novel geometric insight (Section 3). The weight analysis is based on a charging scheme that charges the weight of the spanner to empty regions (Section 4).

The lower bound construction is the scaled lattice with basis vectors of weight $\sqrt{\varepsilon}$ and $\frac{1}{\sqrt{\varepsilon}}$ (Section 2); and not the integer lattice $\mathbb{Z}^d$. We analyze the minimum weight of Euclidean $(1+\varepsilon)$-spanners for the integer grid in the plane.

Theorem 2. *For every $n \in \mathbb{N}$, the minimum weight of a $(1+\varepsilon)$-spanner for the $n \times n$ section of the integer lattice is between $\Omega(\varepsilon^{-3/4} n^2)$ and $O(\varepsilon^{-1}\log(\varepsilon^{-1}) \cdot n^2)$.*

When scaled to n points in $[0,1]^2$, the upper bound confirms that the integer lattice does not maximize the weight of Euclidean $(1+\varepsilon)$-spanners.

Corollary 1. *For every $n \in \mathbb{N}$, the minimum weight of a $(1+\varepsilon)$-spanner for n points in a scaled section of the integer grid in $[0,1]^2$ is between $\Omega(\varepsilon^{-3/4}\sqrt{n})$ and $O(\varepsilon^{-1}\log(\varepsilon^{-1})\sqrt{n})$.*

The lower bound is derived from two elementary criteria (the empty ellipse condition and the empty slab condition) for an edge to be present in every $(1+\varepsilon)$-spanner (Section 2). The upper bound is based on analyzing the SparseYao algorithm from Section 3, combined with results from number theory on Farey sequences (Section 5). Closing the gap between the lower and upper bounds in Theorem 2 remains an open problem. Higher dimensional generalizations are also left for future work. In particular, multidimensional variants of Farey sequences are currently not well understood.

Further Related Previous Work. Many algorithms have been developed for constructing $(1+\varepsilon)$-spanners for n points in $\mathbb{R}^d$ [1,8,9,10,13,19,25,27,29], designed for one or more optimization criteria (lightness, sparsity, hop diameter, maximum degree, and running time). A comprehensive survey up to 2007 is in the book by Narasinham and Smid [28]. We briefly review previous constructions pertaining to the *minimum weight* for n points the unit square (i.e., $d=2$). As noted above, the recent worst-case tight bound on the lightness [23] implies that the greedy algorithm returns a $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-2}\|\mathrm{MST}\|) = O(\varepsilon^{-2}\sqrt{n})$.

A classical method for constructing a $(1+\varepsilon)$-spanners uses *well-separated pair decompositions* (*WSPD*) with a hierarchical clustering (e.g., quadtrees); see [20, Chap. 3]. Due to a hierarchy of depth $O(\log n)$, this technique has been adapted broadly to dynamic, kinetic, and reliable spanners [7,8,17,30]. However, the weight of the resulting $(1+\varepsilon)$-spanner for n points in $[0,1]^2$ is $O(\varepsilon^{-3}\sqrt{n}\cdot\log n)$ [17]. The $O(\log n)$ factor is due to the depth of the hierarchy; and it cannot be removed for any spanner with hop-diameter $O(\log n)$ [3,11,32].

Yao-graphs and Θ-graphs are geometric proximity graphs, defined as follows. For a constant $k \geq 3$, consider k cones of aperture $2\pi/k$ around each point $p \in S$; in each cone, connect p to the "closest" point $q \in S$. For Yao-graphs, q minimizes the Euclidean distance $\|pq\|$, and for Θ-graphs q is the point that minimizes the length of the orthogonal projection of pq to the angle bisector of the cone. It is known that both Θ- and Yao-graphs are $(1+\varepsilon)$-spanners for a parameter $k \in \Theta(\varepsilon^{-1})$, and this bound is the best possible [28]. However, if we place $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ equally spaced points on opposite sides of the unit space, then the weight of both graphs with parameter $k = \Theta(\varepsilon^{-1})$ will be $\Theta(\varepsilon^{-1}\, n)$.

Organization. We start with lower bound constructions in the plane (Section 2) as a warm-up exercise. The two elementary geometric criteria build intuition and highlight the significance of $\sqrt{\varepsilon}$ as the ratio between the two axes of an ellipse of all paths of stretch at most $1+\varepsilon$ between the foci. Section 3 presents Algorithm SparseYao and its stretch analysis in the plane. Its weight analysis for n points in $[0,1]^2$ is in Section 4. We analyze the performance of Algorithm SparseYao for the $n \times n$ grid, after a brief review of Feray sequences, in Section 5. We conclude with a selection of open problems in Section 6. The generalization of Algorithm SparseYao and its analysis are sketched in the full paper [35].

2 Lower Bounds in the Plane

We present lower bounds for the minimum weight of a $(1+\varepsilon)$-spanner for the $n \times n$ section of the integer lattice (Section 2.1); and for n points in a unit square $[0,1]^2$ (Section 2.2).

Let $S \subset \mathbb{R}^2$ be a finite point set. We observe two elementary conditions that guarantee that an edge ab is present *in every* $(1+\varepsilon)$-spanner for S. Two points, $a, b \in S$, determine a (closed) line segment $ab = \mathtt{conv}\{a,b\}$; the relative interior of ab is denoted by $\mathrm{int}(ab) = ab \backslash \{a,b\}$. let $\mathcal{E}_{ab}$ denote the ellipse with foci a and b, and great axis of weight $(1+\varepsilon)\|ab\|$, $\mathcal{L}_{ab}$ be the slab bounded by two lines parallel to ab and tangent lines to $\mathcal{E}_{ab}$; see Fig. 1. Note that the width of $\mathcal{L}_{ab}$ equals the minor axes of $\mathcal{E}_{ab}$, which is $((1+\varepsilon)^2 - 1^2)^{1/2}\|ab\| = (2\varepsilon + \varepsilon^2)^{1/2}\|ab\| > \sqrt{2\varepsilon}\|ab\|$.

- **Empty ellipse condition**: $S \cap \mathcal{E}_{ab} = \{a,b\}$.
- **Empty slab condition**: $S \cap \mathrm{int}(ab) = \emptyset$ and all points in $S \cap \mathcal{L}_{ab}$ are on the line ab.

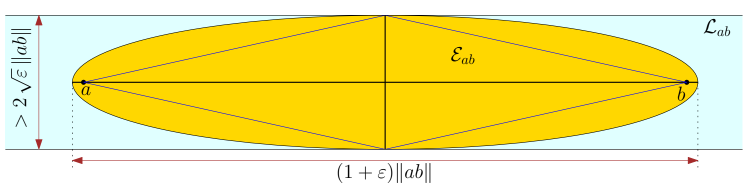

Fig. 1. The ellipse $\mathcal{E}_{ab}$ with foci a and b, and great axis of weight $(1+\varepsilon)\|ab\|$.

Observation 1. *Let $S \subset \mathbb{R}^2$, $G = (S, E)$ a $(1+\varepsilon)$-spanner for S, and $a, b \in S$.*

1. *If ab meets the empty ellipse condition, then $ab \in E$.*
2. *If S is a section of $\mathbb{Z}^2$, $\varepsilon < 1$, and ab meets the empty slab condition, then $ab \in E$.*

Proof. The ellipse $\mathcal{E}_{ab}$ contains all points $p \in \mathbb{R}^2$ satisfying $\|ap\| + \|pb\| \leq (1+\varepsilon)\|ab\|$. Thus, by the triangle inequality, $\mathcal{E}_{ab}$ contains every ab-path of weight at most $(1+\varepsilon)\|ab\|$. The empty ellipse condition implies that such a path cannot have interior vertices.

If S is the integer lattice, then $S \cap \mathrm{int}(ab) = \emptyset$ implies that $\overrightarrow{ab}$ is a primitive vector (i.e., the x- and y-coordinates of $\overrightarrow{ab}$ are relatively prime), hence the distance between any two lattice points along the line ab is at least $\|ab\|$. Given that $\mathcal{E}_{ab} \subset \mathcal{L}_{ab}$, the empty slab condition now implies the empty ellipse condition. $\square$

2.1 Lower Bounds for the Grid

Lemma 1. *For every $n \in \mathbb{N}$ with $n \geq 2\,\varepsilon^{-1/4}$, the weight of every $(1+\varepsilon)$-spanner for the $n \times n$ section of the integer lattice is $\Omega(\varepsilon^{-3/4}n^2)$.*

Proof. Let $S = \{(s_1, s_2) \in \mathbb{Z}^2 : 0 \leq s_1, s_2 < n\}$ and $A = \{(a_1, a_2) \in \mathbb{Z}^2 : 0 \leq a_1, a_2 < \lceil \varepsilon^{-1/4} \rceil / 2\}$. Denote the origin by $o = (0,0)$. For every grid point $a \in A$, we have $\|oa\| \leq \varepsilon^{-1/4}/\sqrt{2}$. A vector $\overrightarrow{oa}$ is *primitive* if $a = (a_1, a_2)$ and $\gcd(a_1, a_2) = 1$. We show that every primitive vector $\overrightarrow{oa}$ with $a \in A$ satisfies the empty slab condition. It is clear that $S \cap \mathrm{int}(oa) = \emptyset$. Suppose that $s \in S$ but it is not on the line spanned by oa. By Pick's theorem, $\mathrm{area}(\Delta(oas)) \geq \frac{1}{2}$. Consequently, the distance between s and the line oa is at least $\|oa\|^{-1} \geq \sqrt{2} \cdot \varepsilon^{1/4} \geq 2\,\varepsilon^{1/2}\,\|oa\|$; and so $s \notin \mathcal{L}_{oa}$, as claimed.

By elementary number theory, $\overrightarrow{oa}$ is primitive for $\Theta(|A|)$ points $a \in A$. Indeed, every $a_1 \in \mathbb{N}$ is relatively prime to $N\varphi(a_1)/a_1$ integers in every interval of length N, where $\varphi(.)$ is Euler totient function, and $\varphi(a_1) = \Theta(a_1)$. Consequently, the total weight of primitive vectors $\overrightarrow{oa}$, $a \in A$, is $\Theta(|A| \cdot \varepsilon^{-1/4}) = \Theta(\varepsilon^{-3/4})$.

The primitive edges oa, $a \in A$, form a star centered at the origin. The translates of this star to other points $s \in S$, with $0 \leq s_1, s_2 \leq \frac{n}{2} \leq n - \lceil \varepsilon^{-1/4} \rceil$ are present in every $(1+\varepsilon)$-spanner for S. As every edge is part of at most two such stars, summation over $\Theta(n^2)$ stars yields a lower bound of $\Omega(\varepsilon^{-3/4}n^2)$. $\square$

Remark 1. The lower bound in Lemma 1 derives from the total weight of primitive vectors $\overrightarrow{oa}$ with $\|oa\| \leq O(\varepsilon^{-1/4})$, which satisfy the empty slab condition. There are additional primitive vectors that satisfy the empty ellipse condition (e.g., $\overrightarrow{oa}$ with $a = (1, a_2)$ for all $|a_2| < \varepsilon^{-1/3}$). However, it is unclear how to account for all vectors satisfying the empty ellipse condition, and whether their overall weight would improve the lower bound in Lemma 1.

Remark 2. The empty ellipse and empty slab conditions each imply that an edge *must* be present in every $(1+\varepsilon)$-spanner for S. It is unclear how the total weight of such "must have" edges compare to the the minimum weight of a $(1+\varepsilon)$-spanner.

2.2 Lower Bounds in the Unit Square

Lemma 2. *For every $n \in \mathbb{N}$ and $\varepsilon \in (0,1]$, there exists a set S of n points in $[0,1]$ such that every $(1+\varepsilon)$-spanner for S has weight $\Omega(\varepsilon^{-3/2}\sqrt{n})$.*

Proof. First let S_0 be a set of $2m$ points, where $m = \lfloor \varepsilon^{-1}/2 \rfloor$, with m equally spaced points on two opposite sides of a unit square. By the empty ellipse property, every $(1+\varepsilon)$-spanner for S_0 contains a complete bipartite graph $K_{m,m}$. The weight of each edge of $K_{m,m}$ is between 1 and $\sqrt{2}$, and so the weight of every $(1+\varepsilon)$-spanner for S_0 is $\Omega(\varepsilon^{-2})$.

For $n \geq \varepsilon^{-1}$, consider an $\lfloor \sqrt{\varepsilon n} \rfloor \times \lfloor \sqrt{\varepsilon n} \rfloor$ grid of unit squares, and insert a translated copy of S_0 in each unit square. Let S be the union of these $\Theta(\varepsilon n)$ copies of S_0; and note that $|S| = \Theta(n)$. A $(1+\varepsilon)$-spanner for each copy of S_0 still requires a complete bipartite graph of weight $\Omega(\varepsilon^{-2})$. Overall, the weight of every $(1+\varepsilon)$-spanner for S is $\Omega(\varepsilon^{-1} n)$.

Finally, scale S down by a factor of $\lfloor \sqrt{\varepsilon n} \rfloor$ so that it fits in a unit square. The weight of every edge scales by the same factor, and the weight of a $(1+\varepsilon)$-spanner for the resulting n points in $[0,1]^2$ is $\Omega(\varepsilon^{-3/2}\sqrt{n})$, as claimed. □

Remark 3. The points in the lower bound construction above lie on $O(\sqrt{\varepsilon n})$ axis-parallel lines in $[0,1]^2$, and so the weight of their MST is $O(\sqrt{\varepsilon n})$. Recall that the lightness of the greedy $(1+\varepsilon)$-spanner is $O(\varepsilon^{-d}\log \varepsilon^{-1})$ [23]. For $d = 2$, it yields a $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-2}\log \varepsilon^{-1}) \cdot \|\mathrm{MST}(S)\| = O(\varepsilon^{-3/2}\log(\varepsilon^{-1})\sqrt{n})$.

3 Spanner Algorithm: Sparse Yao-Graphs

Let S be a set of n points in the plane and $\varepsilon \in (0, \frac{1}{9})$. As noted above, the Yao-graph $Y_k(S)$ with $k = \Theta(\varepsilon^{-1})$ cones per vertex is a $(1+\varepsilon)$-spanner for S. We describe an new algorithm, SparseYao(S, ε), that computes a subgraph of a Yao-graph $Y_k(S)$ (Section 3.1); and show that it returns a $(1+\varepsilon)$-spanner for S (Section 3.2). Later, we use this algorithm for n points in the unit square (Section 4; and for an $n \times n$ section of the integer lattice (Section 5). Our algorithm starts with a Yao-graph that is a $(1+\frac{\varepsilon}{2})$-spanner, in order to leave room for minor loss in the stretch factor due to sparsification. The basic idea is that instead of cones of aperture $2\pi/k = \Theta(\varepsilon)$, cones of much larger aperture $\Theta(\sqrt{\varepsilon})$ suffice in some cases. (This is idea is flashed out in Section 3.2). The angle $\sqrt{\varepsilon}$ then allows us to charge the weight of the resulting spanner to the area of empty regions (specifically, to an empty section of a cone) in Section 4.

3.1 Sparse Yao-Graph Algorithm

We present an algorithm that computes a subgraph of a Yao-graph for S. It starts with cones of aperture $\Theta(\sqrt{\varepsilon})$, and refines them to cones of aperture $\Theta(\varepsilon^{-1})$. We connect each point $p \in S$ to the closest points in the larger cones, and use the smaller cones only when "necessary." To specify when exactly the smaller cones are used, we define two geometric regions that will also play crucial roles in the stretch and weight analyses.

Definitions. Let $p, q \in S$ be distinct points; refer to Fig. 2. Let $A(p,q)$ be the line segment of weight $\frac{\sqrt{\varepsilon}}{2}\|pq\|$ on the line pq with one endpoint at p but interior-disjoint from the ray $\overrightarrow{pq}$; and $\widehat{A}(p,q)$ the set of points in $\mathbb{R}^2$ within distance

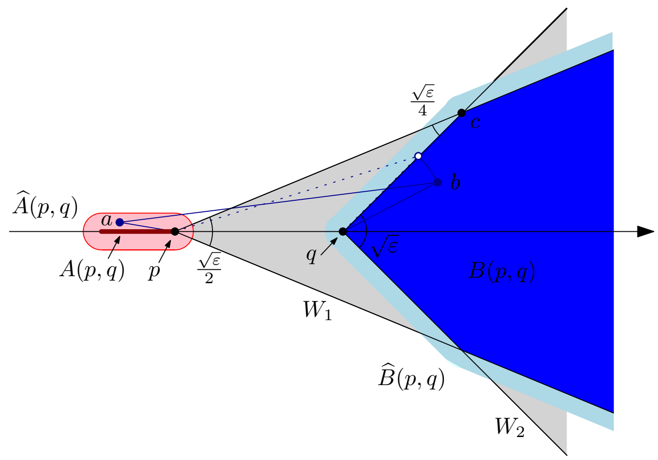

Fig. 2. Wedges W_1 and W_2, line segment $A(p,q)$, and regions $\widehat{A}(p,q)$, $B(p,q)$, and $\widehat{B}(p,q)$ for $p,q \in S$.

$\frac{\varepsilon}{16}\,\|pq\|$ from $A(p,q)$. Let W_1 be the cone with apex p, aperture $\frac{1}{2}\cdot\sqrt{\varepsilon}$, and symmetry axis $\overrightarrow{pq}$; and let W_2 be the cone with apex q, aperture $\sqrt{\varepsilon}$, and symmetry axis $\overrightarrow{pq}$. Let $B(p,q) = W_1 \cap W_2$. Finally, let $\widehat{B}(p,q)$ be the set of points in $\mathbb{R}^2$ within distance at most $\frac{\varepsilon}{8}\,\|pq\|$ from $B(p,q)$.

We show below (cf. Lemma 3) that if we add edge pq to the spanner, then we do not need any of the edges ab with $a \in \widehat{A}(p,q)$ and $b \in \widehat{B}(p,q)$. We can now present our algorithm.

Algorithm SparseYao(S,ε). Input: a set $S \subset \mathbb{R}^2$ of n points, and $\varepsilon \in (0,\frac{1}{9})$.

Preprocessing Phase: Yao-graphs. Subdivide $\mathbb{R}^2$ into $k := \lceil 16\,\pi/\sqrt{\varepsilon}\rceil$ congruent cones of aperture $2\pi/k \leq \frac{1}{8}\cdot\sqrt{\varepsilon}$ with apex at the origin, denoted $C_1,\ldots,C_k$. For $i \in \{1,\ldots,k\}$, let $\overrightarrow{r}_i$ be the symmetry axis of C_i, directed from the origin towards the interior of C_i. For each $i \in \{1,\ldots,k\}$, subdivide C_i into k congruent cones of aperture $2\pi/k^2 \leq \varepsilon/8$, denoted $C_{i,1},\ldots,C_{i,k}$; see Fig. 3. For each point $s \in S$, let $C_i(s)$ and $C_{i,j}(s)$, resp., be the translates of cones C_i and $C_{i,j}$ to apex s.

For all $s \in S$ and $i \in \{1,\ldots,k\}$, let $q_i(s)$ be a closest point to s in $C_i(s)\cap(S\setminus\{s\})$; and for all $j \in \{1,\ldots,k\}$, let $q_{i,j}(s)$ be a closest point in $C_{i,j}(s)\cap(S\setminus\{s\})$; if such points exist. For each $i \in \{1,\ldots,k\}$, let L_i be the list of all ordered pairs $(s,q_i(s))$ sorted in decreasing order of the orthogonal projection of s to the directed line $\overrightarrow{r}_i$; ties are broken arbitrarily.

Main Phase: Computing a Spanner. Initialize an empty graph $G=(S,E)$ with $E := \emptyset$.

1. For all $i \in \{1,\ldots,k\}$, do:
   - While the list L_i is nonempty, do:

   (a) Let (p, q) be the first ordered pair in L_i.
   (b) Add (the unordered edge) pq to E.
   (c) For all $i' \in \{i-2, \ldots, i+2\}$ and $j \in \{1, \ldots, k\}$, do:
   If $\|pq_i(p)\| \leq \|pq_{i',j}(p)\|$ and $q_{i',j}(p) \notin B(p, q)$, then add $pq_{i',j}(p)$ to E.
   (d) For all $s \in \widehat{A}(p, q)$, including $s = p$, delete the pair $(s, q_i(s))$ from L_i.
2. Return $G = (S, E)$.

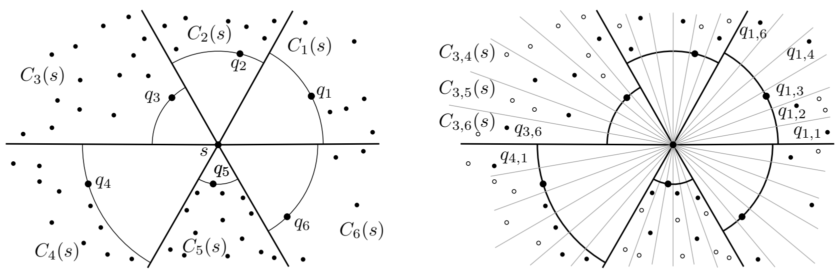

Fig. 3. Cones $C_i(s)$ and $C_{i,j}(s)$ for a point $s \in S$, with $k = 6$.

It is clear that the runtime of Algorithm SparseYao is polynomial in n in the RAM model of computation. In particular, the runtime is dominated preprocessing phase that constructs the Yao-graph with $O(\varepsilon^{-1}n)$ edges: finding the closest points $q_i(s)$ and $q_{i,j}(s)$ is supported by standard range searching data structures [2]. The main phase then computes a subgraph of $Y_{k^2}(S)$ in $O(\varepsilon^{-1}n)$ time. Optimizing the runtime, however, is beyond the scope of this paper.

3.2 Stretch Analysis

In this section, we show that $G = \text{SparseYao}(S, \varepsilon)$ is a $(1+\varepsilon)$-spanner for S. In the preprocessing phase, Algorithm SparseYao computes a Yao-graph with $k^2 = \Theta(\varepsilon^{-1})$ cones. The following lemma justifies that we can omit some of the edges $sq_{i,j}$ from G. In the general case, we have $s = a \in \tilde{A}(p, q)$ and $q_{i,j} = b \in \widehat{B}(p, q)$. For technical reasons, we use a slightly larger neighborhood instead of $\widehat{A}(p, q)$. Let $\tilde{A}(p, q)$ be the set of points in $\mathbb{R}^2$ within distance at most $\frac{\varepsilon}{5}$ from $A(p, q)$.

Lemma 3. *For all $a \in \tilde{A}(p, q)$ and $b \in \widehat{B}(p, q)$, we have*

$$(1+\varepsilon)\|ap\| + \|pq\| + (1+\varepsilon)\|qb\| \leq (1+\varepsilon)\|ab\|. \tag{1}$$

The proof of Lemma 3 is a fairly technical; see the full paper [35]. Next we clarify the relation between $\widehat{A}(p, q)$ and $\tilde{A}(p, q)$.

Lemma 4. *Let $p, q \in S$, and assume that $q \in C_{i',j}(p)$ for some $i, j \in \{1, \ldots, k\}$ and $i' \in \{i-1, i, i+1\}$, where $q_{i',j} = q_{i',j}(p)$ is a closest point to p in $C_{i',j}(p)$. Then $\widehat{A}(p, q_i) \subset \tilde{A}(p, q_{i',j})$.*

Proof. Since the aperture of $C_i(p)$ is $\frac{1}{8} \cdot \sqrt{\varepsilon}$ and $q_i \in C_i(p)$, then $\angle q_i p q_{i',j} \le \frac{1}{4}\sqrt{\varepsilon}$. Since $\|pq_i\| \le \|pq_{i',j}\|$, then $\|A(p, q_i)\| \le \|A(p, q_{i',j})\|$. Consequently, every point in $A(p, q_i)$ is within distance at most $\|A(p, q_i)\| \sin \angle q_i p q_{i',j}) \le \frac{\sqrt{\varepsilon}}{2}\|pq_i\| \cdot \frac{1}{4}\sqrt{\varepsilon} \le \frac{\varepsilon}{8}\|pq_i\|$ from $A(p, q_{i',j})$. By the triangle inequality, the $(\frac{\varepsilon}{16}\|pq_i\|)$-neighborhood of $A(p, q_i)$ is within distance at most $(\frac{\varepsilon}{8} + \frac{\varepsilon}{16})\|pq_i\| < \frac{\varepsilon}{5}\|pq_i\|$ from $A(p, q_{i',j})$. □

The following lemma justifies the role of the regions $\widehat{B}(p, q_i)$. Due to space constraints, its proof is deferred to the full paper [35].

Lemma 5. *Let $p, q \in S$, and assume that $q \in C_{i',j}(p)$ for some $i, j \in \{1, \ldots, k\}$ and $i' \in \{i-1, i, i+1\}$, where $q_{i',j} = q_{i',j}(p)$ is a closest point to p in $C_{i',j}(p)$. If $q \notin B(p, q_i)$ but $q_{i',j} \in B(p, q_i)$, then $q \in \widehat{B}(p, q_i)$.*

Completing the Stretch Analysis. We are now ready to present the stretch analysis for SparseYao(S, ε).

Theorem 3. *For every finite point set $S \subset \mathbb{R}^2$ and $\varepsilon \in (0, \frac{1}{9})$, the graph $G =$ SparseYao(S, ε) is a $(1+\varepsilon)$-spanner.*

Proof. Let S be a set of n points in the plane. Let L_0 be the list of all $\binom{n}{2}$ edges of the complete graph on S sorted by Euclidean weight (ties broken arbitrarily). For $\ell = 1, \ldots, \binom{n}{2}$, let e_ℓ be the ℓ-th edge in L_0, and let $E(\ell) = \{e_1, \ldots, e_\ell\}$. We show the following claim, by induction, for every $\ell = 1, \ldots, \binom{n}{2}$:

Claim. For every edge $ab \in E(\ell)$, $G = (S, E)$ contains an ab-path of weight at most $(1+\varepsilon)\|ab\|$.

For $\ell = 1$, the claim clearly holds, as the shortest edge pq is necessarily the shortest in some cones $C_i(p)$ and $C_{i'}(q)$, as well, and so the algorithm adds pq to E. Assume that $1 < \ell \le \binom{n}{2}$ and the claim holds for $\ell - 1$. If the algorithm added edge e_ℓ to E, then the claim trivially holds for ℓ.

Suppose that $e_\ell \notin E$. Let $e_\ell = pq$, and $q \in C_{i,j}(p)$ for some $i, j \in \{1, \ldots, k\}$. Recall that $q_i = q_i(p)$ is a closest point to p in the cone C_i; and $q_{i,j} = q_{i,j}(p)$ is a closest point to p in the cone $C_{i,j}(p)$. We distinguish between two cases.

(1) The algorithm added the edge pq_i to E. Note that $\|q_i q\| < \|pq\|$ and $\|q_{i,j} q\| < \|pq\|$. By the induction hypothesis, G contains a $q_i q$-path P_i of weight at most $(1+\varepsilon)\|q_i q\|$ and a $q_{i,j} q$-path $P_{i,j}$ of weight at most $(1+\varepsilon)\|q_{i,j} q\|$. If $q \in \widehat{B}(p, q_i)$, then $pq_i + P_i$ is a pq-path of weight at most $(1+\varepsilon)\|pq\|$ by Lemma 3. Otherwise, $q \notin \widehat{B}(p, q_i)$. In this case, $q_{i,j} \notin B(p, q_i)$ by Lemma 5. This means that the algorithm added the edge $pq_{i,j}$ to E. We have $q \in \widehat{B}(p, q_{i,j})$ by Lemma 5, and so $pq_{i,j} + P_{i,j}$ is a pq-path of weight at most $(1+\varepsilon)\|pq\|$ by Lemma 3.

(2) The algorithm did not add the edge pq_i to E. Then the algorithm deleted (p, q_i) from the list L_i in a step in which it added another edge $p'q'_i$ to E. This means that $p \in \widehat{A}(p', q'_i)$, where q'_i is the closest point to p' in the cone $C_i(p')$. As $\mathtt{diam}(A(p_i, q'_i)) < (\sqrt{\varepsilon} + 2 \cdot \frac{\varepsilon}{16})\|p'q'_i\| < \frac{1}{4}\,\|p'q'_i\|$ for $\varepsilon \in (0, \frac{1}{9})$, then $p \in \widehat{A}(p', q'_i)$ implies $\|pp'\| \leq \frac{1}{4}\,\|p'q'_i\|$. Since L_i is sorted by weight, then $\|p'q'_i\| \leq \|pq_i\|$.

Although we have $q \in C_i(p)$, the point q need not be in the cone $C_i(p')$; see Fig. 4. We claim that q lies in the union of three consecutive cones: $q \in C_{i-1}(p') \cup C_i(p') \cup C_{i+1}(p')$. Let $D_i(p')$ be part of the cone $C_i(p')$ outside of the circle of radius $\|p'q'_i\|$ centered at p'. Since $q \in C_i(p)$ and $\|p'q'_i\| < \|pq\|$, then q lies in the translate $D_i(p') + \overrightarrow{p'p}$ of $D_i(p')$. Consider the union of translates:

$$D = D_i(p) + \{\overrightarrow{p'a} : a \in A_{p_i, q'_i}\},$$

and note that $q \in D$. We have $\mathtt{diam}(\widehat{A}(p', q'_i)) \leq (\frac{\sqrt{\varepsilon}}{2} + 2 \cdot \frac{\varepsilon}{16})\|p'q'_i\| < \frac{1}{4}\,\|p'q'_i\|$ for $\varepsilon \in (0, \frac{1}{9})$; and recall that the aperture of $C_i(p')$ is $\gamma := 2\pi/k \leq \frac{1}{8} \cdot \sqrt{\varepsilon}$. We can now approximate $\angle qp'q'_i$ as follows; refer to Fig. 4: $\tan \angle qp'q'_i \leq \|p'q'_i\| \tan\gamma / (\|p'q'_i\| - 2\mathtt{diam}(A(p', q_i))) \leq 2\,\tan\alpha$. Consequently, $\angle qp'q'_i < 2\,\gamma$. It follows that $q \in \bigcup_{i'=i-2}^{i+2} C_{i'}(p')$. We distinguish between two subcases:

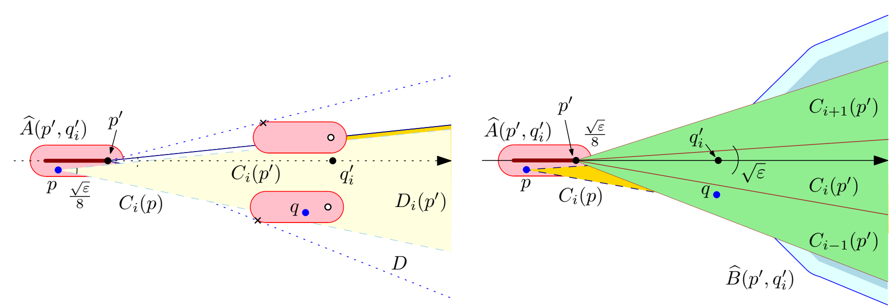

Fig. 4. The relative position of pq and $p'q'_i$. Specifically, $p \in \widehat{A})(p', q'_i)$ and $q \in C_i(p)$. Left: the region $D_i(p')$ and translates of $\widehat{A}(p', q'_i)$ to two critical points of $D_i(p')$. Right: $q \in C_{i-1}(p') \cup C_i(p') \cup C_{i+1}(p')$ and the region $B(p'q'_i)$.

(2a) $q \in \widehat{B}(p', q'_i)$**.** By induction, G contains $(1+\varepsilon)$-paths between p and p', and between q and q'_i. By Lemma 3 (with $a = p$ and $b = q$), the concatenation of these paths and the edge $p'q'_i$ is a pq-path of weight at most $(1+\varepsilon)\|pq\|$.

(2b) $q \notin \widehat{B}(p', q'_i)$. Then $q \in C_{i',j'}(p')$ for some $i' \in \{i-1, i, i+1\}$ and $j' \in \{1, \ldots, k\}$. By Lemma 5, we have $q_{i',j'} \notin B(p', q_i)$. and so the algorithm added the edge $p'q_{i',j'}$, where $q_{i',j'}$ is the closest point to p' in the cone $C_{i',j'}(p')$. We have $p \in \widehat{A}(p', q'_i) \subset \tilde{A}(p', q_{i',j})$ by Lemma 4, and $q \in \widehat{B}(p', q_{i',j})$ by Lemma 5. By induction, G contains $(1+\varepsilon)$-paths between p and p', and between $q_{i',j'}$ and q. The concatenation of these paths and the edge $p'q_{i',j'}$ is a pq-path of weight at most $(1+\varepsilon)\|pq\|$ by Lemma 3. □

4 Spanners in the Unit Square

In this section, we show that for a set $S \subset [0,1]^2$ of n points in the unit square and $\varepsilon \in (0, \frac{1}{9})$, Algorithm SparseYao returns a $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-3/2}\sqrt{n})$ (cf. Theorem 4).

The spanner SparseYao(S, ε) is a subgraph of the Yao-graph with cones of aperture $2\pi/k^2 = O(\varepsilon)$, and so it has $O(\varepsilon^{-1}n)$ edges. Recall that for all $p \in S$ and all $i \in \{1, \ldots, k\}$, there is at most one edge $pq_i(p)$ in G, where $q_i(p)$ is the closest point to p in the cone $C_i(p)$ of aperture $\frac{1}{8}\sqrt{\varepsilon}$. Let

$$F = \{pq_i(p) \in E(G) : p \in S, i \in \{1, \ldots, k\}\}.$$

We first show that the weight of F approximates the weight of the spanner.

Lemma 6. *If Algorithm* SparseYao *adds $pq_i(p)$ and $pq_{i',j}(p)$ to G in the same iteration, then $\|pq_{i',j}(p)\| < 2\,\|pq_i(p)\|$.*

Proof. For short, we write $q_i = q_i(p)$ and $q_{i',j} = q_{i',j}(p)$, where $i \in \{i-1, i, i+1\}$. Since SparseYao added $pq_{i',j}$ to G, then $q_{i',j} \notin B(p, q_i)$. Recall (cf. Fig. 2) that $B(p, q_i) = W_1 \cap W_2$, where W_1 and W_2 are cones centered at p and q_i, resp., with apertures $\frac{1}{2}\sqrt{\varepsilon}$ and $\sqrt{\varepsilon}$. Since the aperture of the cone $C_i(p)$ is $\frac{1}{8}\sqrt{\varepsilon}$, then $C_{i-1}(p) \cup C_i(p) \cup C_{i+1}(p) \subset W_1$, hence $(C_{i-1}(p) \cup C_i(p) \cup C_{i+1}(p)) \setminus B(p, q_i) \subset W_1 \setminus W_2$. The line segment pq_i decomposes $W_1 \setminus W_2$ into two isosceles triangles. By the triangle inequality, the diameter of each isosceles triangle is less than $2\|pq_i\|$. This implies $\|pq'\| < 2\,\|pq_i\|$ for any $q' \in W_1 \setminus W_2$, as claimed. $\square$

Lemma 7. *For $G =$ SparseYao(S, ε), we have $\|G\| = O(\varepsilon^{-1/2}) \cdot \|F\|$.*

Proof. Fix p and $i \in \{1, \ldots, k\}$, let $q_i = q_i(p)$ for short, and suppose $pq_i \in E(G)$. Consider one step of the algorithm that adds the edge pq_i to G, together with up to $3k = \Theta(\varepsilon^{-1/2})$ edges of type $pq_{i',j}$, where $q_{i',j} \notin B(p, q_i)$ and $i' \in \{i-1, i, i+1\}$. By Lemma 6, $\|pq_{i',j}\| < 2\|pq_i\|$. The total weight of all edges $pq_{i',j}$ added to the spanner is

$$\|pq_i(p)\| + \sum_{i'=i-1}^{i+1} \sum_{j=1}^{k} \|pq_{i',j}\| \le \|pq_i\| + 3k \cdot 2\,\|pq_i\| \le O(k\|pq_i\|) \le O(\varepsilon^{-1/2})\|pq_i\|).$$

Summation over all edges in F yields $\|G\| = O(\varepsilon^{-1/2}) \cdot \|F\|$. $\square$

It remains to show that $\|F\| \le O(\varepsilon^{-1}\sqrt{n})$. For $i = 1, \ldots, k$, let $F_i = \{pq_i(p) \in E(G) : p \in S\}$, that is, the set of edges in G between points p and the closest point $q_i(p)$ in cone $C_i(p)$ of aperture $\sqrt{\varepsilon}$. We prove that $\|F_i\| \le O(\varepsilon^{-1/2}\sqrt{n})$ (in the full version of this paper citefull). Since $k = \Theta(\varepsilon^{-1/2})$ this implies the following.

Theorem 4. *For every set of n points in $[0,1]^2$ and every $\varepsilon > 0$, Algorithm* SparseYao *returns a Euclidean $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-3/2}\sqrt{n})$.*

Proof. Let $G =$ SparseYao(S, ε), and define $F \subset E(G)$ and $F_1, \ldots, F_k$ as above. We prove $\|F\| = \sum_{i=1}^{k} \|F_i\| = O(k\,\varepsilon^{-1/2}\sqrt{n}) = O(\varepsilon^{-1}\sqrt{n})$ in the full paper [35]. Now Lemma 7 yields $\|G\| \le O(\varepsilon^{-1/2}) \cdot (\|F\| + \sqrt{2}) \le O(\varepsilon^{-3/2}\sqrt{n})$. $\square$

5 Spanners for the Integer Grid

Two points in the integer lattice $p, q \in \mathbb{Z}^2$ are *visible* if the line segment pq does not pass through any lattice point. An integer point $(i,j) \in \mathbb{Z}^2$ is visible from the origin $(0,0)$ if i and j are relatively prime, that is, $\gcd(i,j)=1$. The *slope* of a segment between $(0,0)$ and (i,j) is j/i. For every $n \in \mathbb{N}$, the *Farey set of order* n, $F_n = \left\{\frac{a}{b} : 0 \leq a \leq b \leq n\right\}$, is the set of slopes of the lines spanned by the origin and lattice points $(b,a) \in [0,n]^2$ with $a \leq b$. The *Farey sequence* is the sequence of elements in F_n in increasing order. Note that $F_n \subset [0,1]$. Farey sets and sequences have fascinating properties, and the distribution of F_n, as $n \to \infty$ is not fully understood [12,16,22,26].

The key result we need is a bound on the average distance to a Farey set F_n. For every $x \in [0,1]$, let

$$\rho_n(x) = \min_{\frac{p}{q} \in F_n} \left| \frac{p}{q} - x \right|$$

denote the distance between x and the Farey set F_n. Kargaev and Zhigljavsky [21] proved that

$$\int_0^1 \rho_n(x)\, dx = \frac{3}{\pi^2}\, \frac{\ln n}{n^2} + O\left(\frac{1}{n^2}\right), \qquad \text{as} \qquad n \to \infty. \tag{2}$$

In the full paper [35], we use (2) to prove the following.

Theorem 5. *Let S be the $n \times n$ section of the integer lattice for some positive integer n. Then the graph $G = \text{SparseYao}(S, \varepsilon)$ has weight $O(\varepsilon^{-1}\log(\varepsilon^{-1}) \cdot n^2)$.*

The combination of Lemma 1 and Theorem 5 establishes Theorem 2.

6 Outlook

Our SparseYao algorithm combines features of Yao-graphs and greedy spanners. It remains an open problem whether the celebrated greedy algorithm [4] always returns a $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-3/2}\sqrt{n})$ for n points in the unit square (and $O(\varepsilon^{(1-d^2)/d} n^{(d-1)/d})$ for n points in $[0,1]^d$). The analysis of the greedy algorithm is known to be notoriously difficult [15,23]. It is also an open problem whether SparseYao or the greedy algorithm achieves an approximation ratio better than the tight lightness bound of $O(\varepsilon^{-d})$ for n points in $\mathbb{R}^d$ (where the approximation ratio compares the weight of the output with the instance-optimal weight of a $(1+\varepsilon)$-spanner).

All results in this paper pertain to Euclidean spaces. Generalizations to L_p-norms for $p \geq 1$ (or Minkowski norms with respect to a centrally symmetric convex body in $\mathbb{R}^d$) would be of interest. It is unclear whether some or all of the machinery developed here generalizes to other norms. Finally, we note that Steiner points can substantially improve the weight of a $(1+\varepsilon)$-spanner in Euclidean space [5,23,24]. It is left for future work to study the minimum weight of a Euclidean Steiner $(1+\varepsilon)$-spanner for n points in the unit square $[0,1]^2$ (or unit cube $[0,1]^d$); and for an $n \times n$ section of the integer lattice.

References

1. Abu-Affash, A.K., Bar-On, G., Carmi, P.: δ-greedy t-spanner. Computational Geometry **100**, 101807 (2022). https://doi.org/10.1016/j.comgeo.2021.101807
2. Agarwal, P.K.: Range searching. In: Goodman, J.E., O'Rourke, J., Tóth, C.D. (eds.) Handbook of Discrete and Computational Geometry, chap. 40, pp. 1057–1092. CRC Press, Boca Raton, FL, 3 edn. (2017)
3. Agarwal, P.K., Wang, Y., Yin, P.: Lower bound for sparse Euclidean spanners. In: Proc. 16th ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 670–671 (2005), http://dl.acm.org/citation.cfm?id=1070432.1070525
4. Althöfer, I., Das, G., Dobkin, D.P., Joseph, D., Soares, J.: On sparse spanners of weighted graphs. Discret. Comput. Geom. **9**, 81–100 (1993). https://doi.org/10.1007/BF02189308
5. Bhore, S., Tóth, C.D.: Light Euclidean Steiner spanners in the plane. In: Proc. 37th Symposium on Computational Geometry (SoCG). LIPIcs, vol. 189, pp. 15:1–15:17. Schloss Dagstuhl (2021). https://doi.org/10.4230/LIPIcs.SoCG.2021.15
6. Borradaile, G., Le, H., Wulff-Nilsen, C.: Greedy spanners are optimal in doubling metrics. In: Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 2371–2379 (2019). https://doi.org/10.1137/1.9781611975482.145
7. Buchin, K., Har-Peled, S., Oláh, D.: A spanner for the day after. Discret. Comput. Geom. **64**(4), 1167–1191 (2020). https://doi.org/10.1007/s00454-020-00228-6
8. Chan, T.M., Har-Peled, S., Jones, M.: On locality-sensitive orderings and their applications. SIAM J. Comput. **49**(3), 583–600 (2020). https://doi.org/10.1137/19M1246493
9. Das, G., Heffernan, P.J., Narasimhan, G.: Optimally sparse spanners in 3-dimensional Euclidean space. In: Proc. 9th Symposium on Computational Geometry (SoCG). pp. 53–62 (1993). https://doi.org/10.1145/160985.160998
10. Das, G., Narasimhan, G., Salowe, J.S.: A new way to weigh malnourished Euclidean graphs. In: Proc. 6th ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 215–222 (1995), http://dl.acm.org/citation.cfm?id=313651.313697
11. Dinitz, Y., Elkin, M., Solomon, S.: Low-light trees, and tight lower bounds for Euclidean spanners. Discret. Comput. Geom. **43**(4), 736–783 (2010). https://doi.org/10.1007/s00454-009-9230-y
12. Dress, F.: Discrépance des suites de Farey. J. Théor. Nombres Bordeaux **11**(2), 345–367 (1999)
13. Elkin, M., Solomon, S.: Optimal Euclidean spanners: Really short, thin, and lanky. J. ACM **62**(5), 35:1–35:45 (2015). https://doi.org/10.1145/2819008
14. Few, L.: The shortest path and the shortest road through n points. Mathematika **2**(2), 141–144 (1955). https://doi.org/10.1112/S0025579300000784
15. Filtser, A., Solomon, S.: The greedy spanner is existentially optimal. SIAM J. Comput. **49**(2), 429–447 (2020). https://doi.org/10.1137/18M1210678
16. Franel, J.: Les suites de Farey et les problemes des nombres premiers. Gottinger Nachrichten pp. 198–201 (1924)
17. Gao, J., Guibas, L.J., Nguyen, A.: Deformable spanners and applications. Comput. Geom. **35**(1-2), 2–19 (2006). https://doi.org/10.1016/j.comgeo.2005.10.001
18. Gottlieb, L.: A light metric spanner. In: Proc. 56th IEEE Symposium on Foundations of Computer Science (FOCS). pp. 759–772 (2015). https://doi.org/10.1109/FOCS.2015.52
19. Gudmundsson, J., Levcopoulos, C., Narasimhan, G.: Fast greedy algorithms for constructing sparse geometric spanners. SIAM J. Comput. **31**(5), 1479–1500 (2002). https://doi.org/10.1137/S0097539700382947

20. Har-Peled, S.: Geometric Approximation Algorithms, Mathematics Surveys and Monographs, vol. 173. AMS (2011)
21. Kargaev, P., Zhigljavsky, A.: Approximation of real numbers by rationals: Some metric theorems. Journal of Number Theory **61**, 209–225 (1996). https://doi.org/10.1006/jnth.1996.0145
22. Landau, E.: Bemerkungen zu der vorstehenden Abhandlung von Herrn Franel. Göttinger Nachrichten pp. 202–206 (1924), coll. works, vol. 8 (Thales Verlag, Essen)
23. Le, H., Solomon, S.: Truly optimal Euclidean spanners. In: Proc. 60th IEEE Symposium on Foundations of Computer Science (FOCS). pp. 1078–1100. IEEE Computer Society (2019). https://doi.org/10.1109/FOCS.2019.00069
24. Le, H., Solomon, S.: Light Euclidean spanners with Steiner points. In: Proc. 28th European Symposium on Algorithms (ESA). LIPIcs, vol. 173, pp. 67:1–67:22. Schloss Dagstuhl (2020). https://doi.org/10.4230/LIPIcs.ESA.2020.67
25. Le, H., Solomon, S.: Towards a unified theory of light spanners I: fast (yet optimal) constructions. CoRR **abs/2106.15596** (2021), https://arxiv.org/abs/2106.15596
26. Ledoan, A.H.: The discrepancy of Farey series. Acta Mathematica Hungarica **156**, 465–480 (2018). https://doi.org/10.1007/s10474-018-0868-x
27. Levcopoulos, C., Narasimhan, G., Smid, M.H.M.: Improved algorithms for constructing fault-tolerant spanners. Algorithmica **32**(1), 144–156 (2002). https://doi.org/10.1007/s00453-001-0075-x
28. Narasimhan, G., Smid, M.H.M.: Geometric Spanner Networks. Cambridge University Press (2007). https://doi.org/10.1017/CBO9780511546884
29. Rao, S., Smith, W.D.: Approximating geometrical graphs via "spanners" and "banyans". In: Proc. 30th ACM Symposium on the Theory of Computing (STOC). pp. 540–550 (1998). https://doi.org/10.1145/276698.276868
30. Roditty, L.: Fully dynamic geometric spanners. Algorithmica **62**(3-4), 1073–1087 (2012). https://doi.org/10.1007/s00453-011-9504-7
31. Ruppert, J., Seidel, R.: Approximating the d-dimensional complete Euclidean graph. In: Proc. 3rd Canadian Conference on Computational Geometry (CCCG). pp. 207–210 (1991), https://cccg.ca/proceedings/1991/paper50.pdf
32. Solomon, S., Elkin, M.: Balancing degree, diameter, and weight in euclidean spanners. SIAM J. Discret. Math. **28**(3), 1173–1198 (2014). https://doi.org/10.1137/120901295
33. Steele, J.M., Snyder, T.L.: Worst-case growth rates of some classical problems of combinatorial optimization. SIAM J. Comput. **18**(2), 278–287 (1989). https://doi.org/10.1137/0218019
34. Supowit, K.J., Reingold, E.M., Plaisted, D.A.: The travelling salesman problem and minimum matching in the unit square. SIAM J. Comput. **12**(1), 144–156 (1983). https://doi.org/10.1137/0212009
35. Tóth, C.D.: Minimum weight Euclidean $(1+\varepsilon)$-spanners. CoRR **abs/2206.14911** (2022), https://arxiv.org/abs/2206.14911