Online Euclidean Spanners

Sujoy Bhore
Indian Institute of Science Education and Research, Bhopal, India

Csaba D. Tóth
California State University Northridge, Los Angeles, CA, USA
Tufts University, Medford, MA, USA

Abstract

In this paper, we study the online Euclidean spanners problem for points in $\mathbb{R}^d$. Given a set $S$ of $n$ points in $\mathbb{R}^d$, a $t$-spanner on $S$ is a subgraph of the underlying complete graph $G = (S, (\binom{S}{2}))$, that preserves the pairwise Euclidean distances between points in $S$ to within a factor of $t$, that is the stretch factor. Suppose we are given a sequence of $n$ points $(s_1, s_2, \ldots, s_n)$ in $\mathbb{R}^d$, where point $s_i$ is presented in step $i$ for $i = 1, \ldots, n$. The objective of an online algorithm is to maintain a geometric $t$-spanner on $S_i = \{s_1, \ldots, s_i\}$ for each step $i$. The algorithm is allowed to add new edges to the spanner when a new point is presented, but cannot remove any edge from the spanner. The performance of an online algorithm is measured by its competitive ratio, which is the supremum, over all sequences of points, of the ratio between the weight of the spanner constructed by the algorithm and the weight of an optimum spanner. Here the weight of a spanner is the sum of all edge weights.

First, we establish a lower bound of $\Omega(\varepsilon^{-1} \log n / \log \varepsilon^{-1})$ for the competitive ratio of any online $(1 + \varepsilon)$-spanner algorithm, for a sequence of $n$ points in 1-dimension. We show that this bound is tight, and there is an online algorithm that can maintain a $(1 + \varepsilon)$-spanner with competitive ratio $O(\varepsilon^{-1} \log n / \log \varepsilon^{-1})$. Next, we design online algorithms for sequences of points in $\mathbb{R}^d$, for any constant $d \geq 2$, under the $L_2$ norm. We show that previously known incremental algorithms achieve a competitive ratio $O(\varepsilon^{-(d+1)} \log n)$. However, if the algorithm is allowed to use additional points (Steiner points), then it is possible to substantially improve the competitive ratio in terms of $\varepsilon$. We describe an online Steiner $(1 + \varepsilon)$-spanner algorithm with competitive ratio $O(\varepsilon^{(1-d)/2} \log n)$. As a counterpart, we show that the dependence on $n$ cannot be eliminated in dimensions $d \geq 2$. In particular, we prove that any online spanner algorithm for a sequence of $n$ points in $\mathbb{R}^d$ under the $L_2$ norm has competitive ratio $\Omega(f(n))$, where $\lim_{n\to\infty} f(n) = \infty$. Finally, we provide improved lower bounds under the $L_1$ norm: $\Omega(\varepsilon^{-d}/ \log \varepsilon^{-1})$ in the plane and $\Omega(\varepsilon^{-d})$ in $\mathbb{R}^d$ for $d \geq 3$.

2012 ACM Subject Classification Mathematics of computing → Approximation algorithms; Mathematics of computing → Paths and connectivity problems; Theory of computation → Computational geometry

Keywords and phrases Geometric spanner, $(1 + \varepsilon)$-spanner, minimum weight, online algorithm

Digital Object Identifier 10.4230/LIPIcs.ESA.2021.16


Funding Csaba D. Tóth: Research was partially supported by the NSF award DMS-1800734.

1 Introduction

We study the online Euclidean spanners problem for a set of points in $\mathbb{R}^d$. Let $S$ be a set of $n$ points in $\mathbb{R}^d$. A $t$-spanner for a finite set $S$ of points in $\mathbb{R}^d$ is a subgraph of the underlying complete graph $G = (S, (\binom{S}{2}))$, that preserves the pairwise Euclidean distances between points in $S$ to within a factor of $t$, that is the stretch factor. The edge weights of $G$ are the Euclidean distances between the vertices. Chew [22, 23] initiated the study of Euclidean spanners in 1986, and showed that for a set of $n$ points in $\mathbb{R}^2$, there exists a spanner with $O(n)$ edges and constant stretch factor. Since then a large body of research has been...
devoted to Euclidean spanners due to its vast applications across domains, such as, topology control in wireless networks [50], efficient regression in metric spaces [31], approximate distance oracles [36], and many others. Moreover, Rao and Smith [48] showed the relevance of Euclidean spanners in the context of other fundamental geometric NP-hard problems, e.g., Euclidean traveling salesman problem and Euclidean minimum Steiner tree problem. Many different spanner construction approaches have been developed for Euclidean spanners over the years, that each found further applications in geometric optimization, such as spanners based on well-separated pair decomposition (WSPD) [17, 35], skip-lists [4], path-greedy and gap-greedy approaches [3, 5], locality-sensitive orderings [21], and more. We refer to the book by Narasimhan and Smid [47] and the survey of Bose and Smid [16] for a summary of results and techniques on Euclidean spanners up to 2013.

Online Spanners. We are given a sequence of \( n \) points \((s_1, s_2, \ldots, s_n)\), where the points are presented one-by-one, i.e., point \( s_i \) is revealed at the step \( i \), and \( S_i = \{ s_1, \ldots, s_i \} \) for \( i = 1, \ldots, n \). The objective of an online algorithm is to maintain a geometric \( t \)-spanner \( G_i \) for \( S_i \) for all \( i \). Importantly, the algorithm is allowed to add edges to the spanner when a new point arrives, however is not allowed to remove any edge from the spanner.

The performance of an online algorithm \( \text{ALG} \) is measured by comparing it to the offline optimum \( \text{OPT} \) using the standard notion of competitive ratio [14, Ch. 1]. The competitive ratio of an online \( t \)-spanner algorithm \( \text{ALG} \) is defined as \( \sup_{\sigma} \frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)} \), where the supremum is taken over all input sequences \( \sigma \), \( \text{OPT}(\sigma) \) is the minimum weight of a \( t \)-spanner for \( \sigma \), and \( \text{ALG}(\sigma) \) denotes the weight of the \( t \)-spanner produced by \( \text{ALG} \) for this input.

Computing a \((1+\varepsilon)\)-spanner of minimum weight for a set \( S \) in Euclidean plane is known to be NP-hard [20]. However, there exists a plethora of constant-factor approximation algorithms for this problem in the offline model; see [3, 25, 26, 48]. Most of these algorithms approximate the parameter lightness (the ratio of the spanner weight to the weight of the Euclidean minimum spanning tree \( \text{MST}(S) \)) of Euclidean spanners, which in turn also approximates the optimum weight of the spanner. We refer to Section 1.1 for a more detailed overview of the parameter lightness.

Minimum spanning trees (MST) on \( n \) points in a metric space, which have no guarantee on the stretch factor, have been studied in the online model. It is not difficult to show that a greedy algorithm achieves a competitive ratio \( \Theta(\log n) \). The online Steiner tree problem was studied by Imase and Waxman [39], who proved \( \Theta(\log n) \)-competitiveness for the problem. Later, Alon and Azar [2] studied minimum Steiner trees for points in the Euclidean plane, and proved a lower bound \( \Omega(\log n / \log \log n) \) for the competitive ratio. Their result was the first to analyse the impact of Steiner points on a geometric network problem in the online setting. Several algorithms were proposed over the years for the online Steiner Tree and Steiner forest problems, on graphs in both weighted and unweighted settings; see [1, 6, 10, 37, 46].

Online Steiner Spanners. An important variant of online spanners is when it is allowed to use auxiliary points (Steiner points) which are not part of input sequence of points. It turns out that Steiner points allow for substantial improvements over the bounds on the sparsity and lightness of Euclidean spanners in the offline settings; see [12, 13, 42, 43]. In the geometric setting, an online algorithm is allowed to add Steiner points and subdivide existing edges with Steiner points at each time step. (This modeling decision has twofold justification: It accurately models physical networks such as roads, canals, or power lines, and from the theoretical perspective, it is hard to tell whether an online algorithm introduced a large number of Steiner points when it created an edge/path in the first place). However, the spanner must achieve the given stretch factor only for the input point pairs.
S. Bhore and C. D. Tóth

1.1 Related Work

Dynamic Spanners. In applications, the data (modeled as points in $\mathbb{R}^d$) changes over time, as new cities emerge, new wireless antennas are built, and users turn their wireless devices on or off. Dynamic models aim to maintain a geometric $t$-spanners for a dynamically changing point set $S$; in a restricted insert-only model, the input consists of a sequence of point insertions. In the dynamic model, the objective is design algorithms and data structures that minimize the worst-case update time needed to maintain a $t$-spanner for $S$ over all steps, regardless of its weight, sparsity, or lightness. Notice that dynamic algorithms are allowed to add or delete edges in each step, while online algorithms cannot delete edges. However, if a dynamic (or dynamic insert-only) algorithm always adds edges for a sequence of points insertions, it is also an online algorithm, and one can analyze its competitive ratio.

Arya et al. [4] designed a randomized incremental algorithm for $n$ points in $\mathbb{R}^d$, where the points are inserted in a random order, and maintains a $t$-spanner of $O(n)$ size and $O(\log n)$ diameter. Their algorithm can also handle random insertions and deletions in $O(\log^2 n \log \log n)$ expected amortized update time. Later, Bose et al. [15] presented an insert-only algorithm to maintain a $t$-spanner of $O(n)$ size and $O(\log n)$ diameter in $\mathbb{R}^d$. Fischer and Har-Peled [29] used dynamic compressed quadtrees to maintain a WSPD-based $(1+\varepsilon)$-spanner for $n$ points in $\mathbb{R}^d$ in expected $O(\log n + \log \varepsilon^{-1})$ update time. Their algorithm works under the online model, too, however, they have not analyzed the weight of the resulting spanner. Gao et al. [30] used hierarchical clustering for dynamic spanners in $\mathbb{R}^d$. Their DefSpanner algorithm is fully dynamic with $O(\log \Delta)$ update time, where $\Delta$ is the spread$^1$ of the set $S$. They maintain a $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-(d+1)}\|\text{MST}(S)\| \log \Delta)$, and for a sequence of point insertions, DefSpanner only adds edges. As $\text{OPT} \geq \|\text{MST}(S)\|$, DefSpanner can serve as an online algorithm with competitive ratio $O(\varepsilon^{-(d+1)} \log \Delta)$.

---

$^1$ The spread of a finite set $S$ in a metric space is the ratio of the maximum pairwise distance to the minimum pairwise distance of points in $S$; and $\log \Delta \geq \Omega(\log n)$ in doubling dimensions.
Gottlieb and Roditty [32] studied dynamic spanners in more general settings. For every set of $n$ points in a metric space of bounded doubling dimension\(^2\), they constructed a $(1 + \varepsilon)$-spanner whose maximum degree is $O(1)$ and that can be maintained under insertions and deletions in $O(\log n)$ amortized update time per operation. Later, Roditty [49] designed fully dynamic geometric $t$-spanners with optimal $O(\log n)$ update time for $n$ points in $\mathbb{R}^d$. Very recently, Chan et al. [21] introduced locality sensitive orderings in $\mathbb{R}^d$, which has applications in several proximity problems, including spanners. They obtained a fully dynamic data structure for maintaining a $(1 + \varepsilon)$-spanners in Euclidean space with logarithmic update time and linearly many edges. However, the spanner weight has not been analyzed for any of these constructions. Dynamic spanners have been subject to investigation in abstract graphs, as well. See [8, 9, 11] for some recent progress on dynamic graph spanners.

**Lightness and sparsity** are two natural parameters for Euclidean spanners. For a set $S$ of points in $\mathbb{R}^d$, the lightness is the ratio of the spanner weight (i.e., the sum of all edge weights) to the weight of the Euclidean minimum spanning tree $\text{MST}(S)$. It is known that greedy-spanner ([3]) has constant lightness; see [25, 26]. Later, Rao and Smith [48] in their seminal work, showed that the greedy spanner has lightness $\varepsilon^{-O(d)}$ in $\mathbb{R}^d$ for every constant $d$, and asked what is the best possible constant in the exponent. Then, the sparsity of a spanner on $S$ is the ratio of its size to the size of a spanning tree. Classical results [23, 24, 40, 53] show that when the dimension $d \in \mathbb{N}$ and $\varepsilon > 0$ are constant, every set $S$ of $n$ points in $d$-space admits an $(1 + \varepsilon)$-spanners with $O(n)$ edges and weight proportional to that of the Euclidean MST of $S$.

**Dependence on $\varepsilon > 0$ for constant dimension $d$.** The dependence of the lightness and sparsity on $\varepsilon > 0$ for constant $d \in \mathbb{N}$ has been studied only recently. Le and Solomon [42] constructed, for every $\varepsilon > 0$ and constant $d \in \mathbb{N}$, a set $S$ of $n$ points in $\mathbb{R}^d$ for which any $(1 + \varepsilon)$-spanner must have lightness $\Omega(\varepsilon^{-d})$ and sparsity $\Omega(\varepsilon^{-d+1})$, whenever $\varepsilon = \Omega(n^{-1/(d-1)})$. Moreover, they showed that the greedy $(1 + \varepsilon)$-spanner in $\mathbb{R}^d$ has lightness $O(\varepsilon^{-d} \log \varepsilon^{-1})$. In fact, Le and Solomon [42] noticed that Steiner points can substantially improve the bound on the lightness and sparsity of an $(1 + \varepsilon)$-spanner. For minimum sparsity, they gave an upper bound of $O(\varepsilon^{(1-d)/2})$ for $d$-space and a lower bound of $\Omega(\varepsilon^{-1/2} / \log \varepsilon^{-1})$. For minimum lightness, they gave a lower bound of $\Omega(\varepsilon^{-1} / \log \varepsilon^{-1})$, for points in the plane ($d = 2$) [42]. More recently, Bhore and Tóth [13] established a lower bound of $\Omega(\varepsilon^{-d/2})$ for the lightness of Steiner $(1 + \varepsilon)$-spanners in Euclidean $d$-space for all $d \geq 2$. Moreover, for points in the plane, they established an upper bound of $O(\varepsilon^{-1})$ [12].

### 1.2 Our Contributions

We present the main contributions of this paper, and sketch the key technical and conceptual ideas used for establishing these results. (Refer to the technical sections for precise definitions, complete proofs, and additional remarks.)

**Points on a line.** In Section 2 (Theorem 3), we establish a lower bound $\Omega(\varepsilon^{-1} \log n / \log \varepsilon^{-1})$ for the competitive ratio of any online algorithm for a sequence of points on the real line. Moreover, we show that this bound is tight. We present an online algorithm that maintains a $(1 + \varepsilon)$-spanner with competitive ratio $O(\varepsilon^{-1} \log n / \log \varepsilon^{-1})$.

---

\(^2\) A metric is said to be of a **constant doubling dimension** if a ball with radius $r$ can be covered by at most a constant number of balls of radius $r/2$. 
Our online algorithm is a 1-dimensional instantiation of hierarchical clustering, which
was used by Roditty [49] for dynamical spanners in doubling metrics. When a new point
\( s_i \) is “close” to a previous point \( s_j \), we add \( s_i \) to the “cluster” of \( s_j \), otherwise we open a
new cluster. The key question is to define when \( s_i \) is “close” to a previous point. Instead of
the closest points on the line, we find the shortest edge \( pq \) that contains \( s_i \) in the current
spanner, and say that \( s_i \) is “close” to \( p \) (resp., \( q \)) if \( \|ps_i\| \leq \frac{\epsilon}{4}\|pq\| \) (resp., \( \|qs_i\| \leq \frac{\epsilon}{4}\|pq\| \)).
The algorithm (and its analysis), does not explicitly maintain “clusters,” though. It is easy
to show, by induction, that ALG maintains a \((1 + \epsilon)\)-spanner. The main contribution is a
tight analysis of the competitive ratio. We partition the edges into buckets by weight, where bucket \( E_i \)
contains edges \( e \) of weight \( \epsilon^{-(d+1)} < \|e\| \leq \epsilon^{-d} \). The edges of the spanner will
form a laminar family (any edges are interior-disjoint or one contains the other); and the edge
weight decay by factors of at most \((1 - \frac{\epsilon}{4})\) along the descending paths in the containment
poset. Since \((1 - \frac{\epsilon}{4})^{4/\epsilon} < \frac{1}{2} \), we can show that the total weight of edges in a level decreases
by a factor of \( \frac{1}{2} \) after every \( \lceil \frac{\log n}{\epsilon} \rceil \) levels. Thus, the sum of edge weights in a block of \( \lceil \frac{\log n}{\epsilon} \rceil \) consecutive levels is \( O(\epsilon^{-d+1}\text{OPT}) \). This bound, applied to \( O(\log_{\epsilon^{-1}} n) = O(\log n / \log \epsilon^{-1}) \)
buckets, proves the upper bound. The lower bound construction matches the upper bound
for each block of levels and for each bucket.

**Euclidean \( d \)-space without Steiner points.** In Section 3, we study the online Euclidean
spanners for a sequence of points in \( \mathbb{R}^d \). For constant \( d \geq 2 \) and parameter \( \epsilon > 0 \), we
show that the dynamic algorithm by Fischer and Har-Peled achieves, in the online model,
competitive ratio \( O(\epsilon^{-(d+1)}\log n) \) for \( n \) points in \( \mathbb{R}^d \) (Theorem 4 in Section 3.1), matching
the competitive ratio of DefSpanner by Gao et al. [30, Lemma 3.8].

The new competitive analysis of this algorithm is instrumental for extending the algorithm
and its analysis to online Steiner \((1 + \epsilon)\)-spanners (see below). We briefly describe a key
geometric insight. It is well known that for \( a, b \in \mathbb{R}^d \), any \( ab \)-path of weight at most
\((1 + \epsilon)\|ab\| \) lies in an ellipsoid \( B_{ab} \) with foci \( a \) ans \( b \) and great axes \((1 + \epsilon)\|ab\| \). Summation
over disjoint ellipses gives a lower bound for \( \text{OPT} \). Unfortunately, ellipsoids \( B_{ab} \) for all
pairs \( ab \in S \) may heavily overlap. Recently, Bhore and Tóth [13, Lemma 3] proved that
any \( ab \)-path of weight at most \((1 + \epsilon)\|ab\| \) must contain edge of total weight at least \( \frac{1}{2}\|ab\| \)
that are “near-parallel” to \( ab \) (technically, they make an angle at most \( \epsilon^{1/2} \) with \( ab \)); see
Fig. 4(right). By partitioning the edges of the unknown \( \text{OPT} \) spanner by both directions and
disjoint ellipsoids, we obtain a bound of \( \frac{\text{ALG}}{\text{OPT}} \leq O(\epsilon^{-(d+1)}\log n) \).

**Euclidean \( d \)-space with Steiner points.** When we are allowed to use Steiner points, we can
substantially improve the competitive ratio in terms of \( \epsilon \): We describe an algorithm with
competitive ratio \( O(\epsilon^{(1-d)/2}\log n) \) (Theorem 5 in Section 3.2).

The online Steiner algorithm adds a secondary layer to the non-Steiner algorithm: For
each edge \( ab \) of the non-Steiner spanner \( G_1 \), we maintain a path of weight \((1 + \epsilon)\|ab\| \) with
Steiner points; the stretch factor of the resulting Steiner spanner \( G_2 \) is \((1 + \epsilon)^2 < (1 + 3\epsilon) \).
The key idea is to reduce the weight to maintain buckets of edges of \( G_1 \) that have roughly the same
direction and weight, and are nearby locations; and we construct a common Steiner network
\( N \) for them. Importantly, we can construct a “backbone” of the network \( N \) when the first
edge \( ab \) in a bucket arrives, and we have \( \|N\| \leq O(\epsilon^{(1-d)/2}\|ab\|) \). When subsequent edges
\( a'b' \) in the same bucket arrive, then we can add relatively short “connectors” to \( N \) so that it
also contains an \( a'b' \)-path of weight at most \((1 + \epsilon)\|a'b'\| \). Thus \( N \) can easily accommodate
new paths in the online model. The key technical tool for constructing Steiner networks \( N \)
one for each bucket) is the so-called shallow-light trees, introduced by Awerbuch et al. [7]
and Khuller et al. [41], and optimized in the geometric setting by Elkin and Solomon [28, 52].
As a counterpart, we show (Theorem 7 in Section 4) that the dependence on \( n \) cannot be eliminated in dimensions \( d \geq 2 \). In particular, we prove that any \((1 + \varepsilon)\)-spanner for a sequence of \( n \) points in \( \mathbb{R}^d \), has competitive ratio \( \Omega(f(n)) \) for some function \( f(n) \) with \( \lim_{n \to \infty} f(n) = \infty \). The lower bound construction consists of an adaptive strategy for the adversary in the plane: The adversary recursively maintains a space partition and places points in rounds so that the spanner constructed so far is disjoint from most of the ellipses \( B_{ab} \) that will contain the \( ab \)-paths for pairs of new points \( a, b \). In order to control OPT, the adversary maintains the property that OPT, is an \( x \)-monotone path \( \gamma_i \) after round \( i \). However, this requirement means that any new point must be very close to \( \gamma_i \), and \( S \) will be a set of almost collinear points. The core challenge of the Steiner spanner problem seems to lie in the case of almost collinear points.

**Higher dimensions under the \( L_1\)-norm.** Finally, in the full version of our paper we provide improved lower bounds for points in \( \mathbb{R}^d \) under the \( L_1 \) norm (without Steiner points). We show that for every \( \varepsilon > 0 \), under the \( L_1 \) norm, the competitive ratio of any online \((1 + \varepsilon)\)-spanner algorithm \( \Omega(e^{-2}/\log e^{-1}) \) in \( \mathbb{R}^2 \) and is \( \Omega(e^{-d}) \) in \( \mathbb{R}^d \) for \( d \geq 3 \).

The adversary takes advantage of the non-monotonicity of OPT, mentioned above. In round 1, it presents a point set \( S_1 \cup S_2 \) for which any \((1 + \varepsilon)\)-spanner (without Steiner points) must contain a complete bipartite graph between \( S_1 \) and \( S_2 \); however, the optimal Steiner \((1 + \varepsilon)\)-spanner for \( S_1 \cup S_2 \) has much smaller weight. Then in round 2, the adversary presents all Steiner points \( \tilde{S}_1 \cup \tilde{S}_2 \) of an optimal Steiner \((1 + \varepsilon)\)-spanner for \( S_1 \cup S_2 \). The key insight is that under the \( L_1\)-norm (and for this particular point set), the optimal Steiner spanner for \( S_1 \cup S_2 \) already contains Manhattan paths between any two points in \( S = (S_1 \cup S_2) \cup (\tilde{S}_1 \cup \tilde{S}_2) \), and so it remains the optimum solution (without Steiner points) for the point set \( S \).

We were unable to replicate this phenomenon under the \( L_2\)-norm, where the current best lower bound in \( \mathbb{R}^d \), for all \( d \geq 1 \), derives from the 1-dimensional construction. In particular, it is not sufficient to consider the Steiner ratio for \((1 + \varepsilon)\)-spanners, defined as the supremum ratio between the weight of the minimum \((1 + \varepsilon)\)-spanner and the minimum Steiner \((1 + \varepsilon)\)-spanner of a finite point set in \( \mathbb{R}^d \). Under the \( L_2\)-norm, this ratio is \( \Theta(e^{-1}) \) in the plane and \( \Theta(e^{(1-d)/2}) \) in \( \mathbb{R}^d \) for \( d \geq 3 \) \([12, 42, 44]\). However, an optimal Steiner \((1 + \varepsilon)\)-spanner, need not achieve the desired \( 1 + \varepsilon \) stretch factor for the Steiner points.

## 2 Lower and Upper Bounds for Points on a Line

It is easy to analyze the one-dimensional case as the offline optimum network (OPT) for any set of points in a line is a path from the leftmost point to the rightmost point; the stretch factor of this path is always 1. (In contrast, in 2- and higher dimensions, the optimum \((1 + \varepsilon)\)-spanner is highly dependent on the distribution of points, which in turn may change over time in the online model.)

**Lower bound.** The following adversarial strategy establishes a lower bound \( L(n) = \Omega(e^{-1}) \) for the competitive ratio; refer to Fig. 2 (left). Start with two points \( p_0 = 0 \) and \( q_0 = 1 \). For the first two points, ALG must add a direct edge \( p_0q_0 \). Then the adversary successively places points \( p_i = i \cdot \frac{1}{2^i} \), for \( i = 1, \ldots, n \) so that all points remain in the interval \([0, \frac{1}{2}]\). Thus the number of points is \( n = 2 + \frac{1}{e^{-1}} \). In each round, ALG must add the edge \( p_iq_0 \), otherwise any path between \( p_i \) and \( q_0 \) would have to make a detour via a point in \( \{p_0, \ldots, p_{i-1}\} \), and so it would be longer than \((1 + \varepsilon)\|p_iq_0\| \). Since \( \|p_iq_0\| \geq \frac{1}{2} \), the weight of the network after \( n - 2 \) iterations is at least ALG \( \geq 1 + \frac{1}{2}(n - 2) \geq 1 + \frac{1}{2} [e^{-1}] \). Combined with OPT = 1, this yields a lower bound of \( \Omega(e^{-1}) \) for the competitive ratio.
We note that properties (P1)–(P3) are inherently 1-dimensional, as the edges are intervals in time when edge
otherwise, let $G_{i-1} = ab + as_i + s_ib$. If\[\min\{\|ps_i\|,\|s_iq\|\} > \frac{\epsilon}{2}\|pq\|\], we add both $as_i$ and $s_ib$ to $G_i$, that is, $G_i = G_{i-1} + as_i + s_ib$. Otherwise, let $G_i = G_{i-1} + as_i$ if $\|ps_i\| \leq \|s_iq\|$, or else $G_i = G_{i-1} + s_ib$.

We observe a few properties of $G_i$ that are immediate from the construction: (P1) At the time when edge $e$ is added to $G_i$, then the interior of $e$ does not contain any vertices. (P2) The edges in $G_i$ form a laminar set of intervals (i.e., any two edges are interior-disjoint, or one contains the other). (P3) If $e_1, e_2$ are edges in $G_i$ and $e_2 \subset e_1$, then $\|e_2\| \leq (1 - \frac{\epsilon}{2})\|e_1\|$.

We note that properties (P1)–(P3) are inherently 1-dimensional, as the edges are intervals in $\mathbb{R}$, and they do not seem to generalize to higher dimensions.
Lemma 1. For $i = 1, \ldots, n$, the graph $G_i$ is a $(1 + \varepsilon)$-spanner for $S_i$.

The standard proof (by induction on $n$) is available in the full version of the paper.

Lemma 2. For $i = 1, \ldots, n$, we have $\|G_i\| \leq O(\varepsilon^{-1} \text{OPT}, \log i / \log \varepsilon^{-1})$.

Proof. We may assume w.l.o.g. that $i = n$, and let $\text{OPT} = \text{OPT}_n$ for brevity. Let $E$ be the edge set of $G_n$. The order in which $\text{ALG}$ adds edges to $E$ defines a (precedence) poset on $E$. We partition $E$ by weight as follows: Let $\beta = \varepsilon^{-1}$; and for all $\ell \in \mathbb{Z}$, let $E_{\ell}$ be the set of edges $e \in E$ with $\beta^\ell < \|e\| \leq \beta^{\ell+1}$. Since $\|e\| \leq \text{OPT}$ for all $e \in E$, every edge is in $E_{\ell}$ for some $\ell \leq \log_\beta \text{OPT}$. Furthermore, for all $\ell \leq \log_\beta (\text{OPT}/n^2)$, the edges $e \in E_{\ell}$ have weight $\|e\| \leq \text{OPT}/n^2$, and so the total weight of these edges is less than $\text{OPT}$. It remains to consider $E_{\ell}$ for $\log_\beta (\text{OPT}/n^2) \leq \ell \leq \log_\beta \text{OPT}$, that is, for $O(\log n / \log \varepsilon^{-1})$ values of $\ell$.

Let $pq$ be an edge in $E_{\ell}$ that is not contained in any previous edge in $E_{\ell}$. By property (P2), the edges in $E_{\ell}$ form a laminar family, and so $pq$ does not overlap with any previous edge in $E_{\ell}$; and $pq$ contains any subsequent edge that overlaps with it. Let $E_{\ell}(pq)$ be the set of all edges in $E_{\ell}$ that are contained in $pq$ (including $pq$). We claim that

$$\|E_{\ell}(pq)\| \leq O(\varepsilon^{-1} \|pq\|).$$

Summation over all edges $pq \in E_{\ell}$ that are not contained in previous edges in $E_{\ell}$ implies $\|E_{\ell}\| \leq O(\varepsilon^{-1} \text{OPT})$. Summation over all $\ell \in \mathbb{Z}$ then yields

$$\|E\| = \sum_{\ell \in \mathbb{Z}} \|E_{\ell}\| = \sum_{\ell = \lceil \log_\beta (\text{OPT}/n^2) \rceil}^{\lfloor \log_\beta \text{OPT} \rfloor} \|E_{\ell}\| + O(\text{OPT}) = O(\varepsilon^{-1} \text{OPT} \log_\beta n).$$

To prove (1), consider the containment poset of $E_{\ell}(pq)$. In fact, we represent the poset as a rooted binary tree $T$: The root corresponds to $pq$, and edges $e_1, e_2 \in E_{\ell}(pq)$ are in parent-child relation iff $e_2 \subset e_1$, and there is no edge $e' \in E_{\ell}(pq)$ with $e_2 \subset e' \subset e_1$. Each level of $T$ corresponds to interior-disjoint edges contained in $\|pq\|$, so the sum of weight on each level is at most $\|pq\|$. The total weight of the first $k = \lceil 5\varepsilon^{-1} \rceil$ levels is $O(\varepsilon^{-1} \|pq\|)$.

We claim that the total weight on level $k = \lceil 5\varepsilon^{-1} \rceil$ is at most $\frac{1}{2}\|pq\|$. We distinguish between three types of nodes in the subtree of $T$ between levels 0 and $k$: A branching node has two children, a single-child node has one child, and a leaf has no children (in particular all nodes in level $k$ are considered leaves in this subtree). The nodes (leaves) at level $k$ correspond to interior-disjoint edges $e \subset pq$ with $\|e\| \geq \varepsilon \|pq\|$ by the definition of $E_{\ell}(pq)$. Thus there are at most $\lceil \varepsilon^{-1} \rceil$ nodes at level $k$, hence there are less than $\varepsilon^{-1}$ branching nodes. This implies that for any node $e$ on level $k$, the descending path from the root $pq$ to $e$ contains at least $k - \lceil \varepsilon^{-1} \rceil \geq \lceil 4\varepsilon^{-1} \rceil$ single-child nodes.

For the purpose of bounding the total weight at level $k$, we can modify $T$, by incrementally moving all single-child nodes below all branching nodes as follows. While there is an edge $uv$ in $T$, such that $u$ is a branching node, and its parent $v$ is a single-child node, we suppress $u$ and subdivide the two edges of $T$ below $u$ with new nodes $v_1$ and $v_2$. The weight along the edge $uv$ goes down by a factor of at most $(1 - \frac{1}{4})$ by property (P3); we set the weights in the modified tree such that the same decrease occurs along the edges $uv_1$ and $uv_2$. Then each operation maintains property (P3), and the total weight at level $k$ does not change. When the while loop terminates, we obtain a full binary tree with a chain attached to each leaf. As we argued above, each chain has length $\lfloor 4/\varepsilon \rfloor$ or more. The full binary tree does not necessarily decrease the weight. Along each chain of $\lfloor 4/\varepsilon \rfloor$ or more single-child nodes, the weight is cumulatively multiplied by a factor of at most $(1 - \varepsilon/4)^{\lfloor 4/\varepsilon \rfloor} < \frac{1}{2}$. Overall, the total weight at level $k = \lceil 5\varepsilon^{-1} \rceil$ is at most $\frac{1}{2}\|pq\|$, as claimed.
By induction, for every integer $j \geq 0$, the total weight at level $jk = j[5\varepsilon^{-1}]$ is at most $\|pq\|/2^j$. Consequently, the total weight of a block of $k$ consecutive levels $(jk+1,\ldots,(j+1)k]$ is at most $k\|pq\|/2^j$. Overall, $\|E_t(pq)\| = \sum_{j \geq 0} k\|pq\|/2^j = O(k\|pq\|) = O(\varepsilon^{-1} \|pq\|)$, which completes the proof of (1).

We can summarize the discussion above in the following theorem.

\begin{enumerate}
\item \textbf{Theorem 3.} For every $\varepsilon > 0$, the competitive ratio of any online algorithm for $(1+\varepsilon)$-spanners for a sequence of points on a line is $\Omega(\varepsilon^{-1} \log n/ \log \varepsilon^{-1})$. Moreover, there is an online algorithm that maintains a $(1+\varepsilon)$-spanner with competitive ratio $O(\varepsilon^{-1} \log n/ \log \varepsilon^{-1})$.
\end{enumerate}

3. Upper Bounds for Spanners in $\mathbb{R}^d$ under the $L_2$ Norm

We turn to online $(1+\varepsilon)$-spanners in Euclidean $d$-space for $d \geq 2$. The dynamic algorithm DefSpanner by Gao et al. [30], based on hierarchical clustering, achieves $O(\varepsilon^{-(d+1)} \log n)$ competitive ratio in the online model. In Section 3.1, we recover the same bound with a new analysis, where we refine the hierarchical clustering with a partition of the edges into buckets of similar directions, locations, and weights. In Section 3.2, we extend the new analysis to show that the competitive ratio improves to $O(\varepsilon^{(1-d)/2} \log n)$ if we are allowed to use Steiner points. Our spanner algorithm replaces each bucket of “similar” edges with a Steiner network using grids and shallow-light trees, for up to $O(\varepsilon^{(1-d)/2})$ directions.

Preliminaries. \textit{Well-separated pair-decomposition} (for short, WSPD) of a finite point set $S$ in a metric space is a classical tool for constructing $(1+\varepsilon)$-spanners [19, 34, 47, 51]. It is a collection of pairs $\{(A_i, B_i) : i \in I\}$ such that for all $i \in I$, we have $A_i, B_i \subset S$ and max$\{\text{diam}(A_i), \text{diam}(B_i)\} \leq \varepsilon \text{dist}(A_i, B_i)$; and for every point pair $\{s, t\} \in \binom{S}{2}$, there is a pair $(A_i, B_i)$ such that $A_i$ and $B_i$ each contain precisely one of $s$ and $t$. It was shown by Callahan and Kosaraju [18] that if a graph $G = (S, E)$ contains an edge between arbitrary points in $A_i$ and $B_i$, for all $i \in I$, then $G$ is an $(1+O(\varepsilon))$-spanner for $S$; see also [47, Ch. 9].

Dynamic spanners (including the fully dynamic algorithm by Roditty [49] and DefSpanner by Gao et al. [30]) rely on WSPDs and hierarchical clustering. In $\mathbb{R}^d$, hierarchical clustering can be obtained by classical recursive space partitions such as quadtrees [27, Ch. 14]. Dynamic quadtrees and their variants have been studied extensively, due to their broad range of applications; see [38, Ch. 2]. In general, dynamic quadtrees can handle both point insertion and deletion operations. However, in the context of an online algorithm, where the points are only inserted, note that no cell of the quadtree is ever deleted. We analyse the competitive ratio of the dynamic incremental algorithm by Fischer and Har-Peled [29] that maintains an $(1+\varepsilon)$-spanner for $n$ points in Euclidean $d$-space in expected $O((\log n + \log \varepsilon^{-1}) \varepsilon^{-d} \log n)$ update time. However, they have not analyzed the ratio between the \textit{weight} of the resulting $(1+\varepsilon)$-spanner and the minimum weight of an $(1+\varepsilon)$-spanner.

3.1 Online Algorithm without Steiner Points

\textbf{Online Algorithm.} We briefly review the algorithm in [29] and then analyze the weight. The input is a sequence of points $(s_1, s_2, \ldots)$ in $\mathbb{R}^d$; the set of the first $n$ points is denoted by $S_n = \{s_i : 1 \leq i \leq n\}$. For every $n$, we dynamically maintain a quadtree $T_n$ for $S_n$. Every node of $T_n$ corresponds to a cube. The root of $T_n$, at level 0, corresponds to a cube $Q_0$ of side length $a_0 = \Theta(\text{diam}(S_n))$. At every level $\ell \geq 0$, there are at most $2^{d\ell}$ interior-disjoint cubes, each of side length $a_0/2^\ell$. A cube $Q \in T_n$ is \textit{nonempty} if $Q \cap S_n \neq \emptyset$. For every nonempty cube $Q$, we select an arbitrary representative $s(Q) \in Q \cap S_n$. At each level $\ell$,
let $E_\ell$ be the set of all edges $s(Q_1)s(Q_2)$ for pairs of cubes $\{Q_1, Q_2\}$ on level $\ell$ such that $\frac{c_1n}{\epsilon^{2d}} \leq ||s(Q_1)s(Q_2)|| \leq \frac{c_2n}{\epsilon^{2d}}$ for some constants $0 < c_1 < c_2$ that depend on $d$; see Fig. 4(left).

The algorithm maintains the spanner $G = (S_n, E)$ where $E = \bigcup_{\ell \geq 0} E_\ell$. A classical argument by Callahan and Kosaraju [18] (see also [34, 47, 51]) shows that $G$ is a $(1+\epsilon)$-spanner for $S_n$.

![Figure 4](left: Nonempty squares at level $\ell = 4$ of a quadtree, each with a representative (red dots). Point $s(Q)$ is connected to all other representatives in the annulus between the concentric circles $C_1$ and $C_2$ of radii $c_1/(\epsilon 2^d)$ and $c_2/(\epsilon 2^d)$. Right: Ellipse $B_{ab}$ with foci $a$ and $b$, an $ab$-path of weight $(1+\epsilon)||ab||$. The bold edges make an angle at most $\epsilon^{1/2}$ with $ab$.)

**Theorem 4.** For every constant $d \geq 2$, parameter $\epsilon > 0$, and a sequence of $n \in \mathbb{N}$ points in Euclidean $d$-space, the competitive ratio of the online algorithm above is in $O(\epsilon^{-(d+1)\log n})$.

**Proof.** For the set $S_n$ of the first $n$ points of a sequence in $\mathbb{R}^d$, let $G = (S_n, E)$ be the $(1+\epsilon)$-spanner produced by the online algorithm, and let $G^* = (S_n, E^*)$ be an $(1+\epsilon)$-spanner of minimum weight. We show that $||G||/||G^*|| = O(\epsilon^{-(d+1)\log n})$.

**Short edges.** Note that the weight of every edge in $E_\ell \subset E$ at level $\ell$ is $\Theta(\epsilon^{-1}\text{diam}(S_n)/2^d)$, since it connects representatives at $\Theta(\epsilon^{-1}\text{diam}(S_n)/2^d)$ distance apart. In particular, an edge at any level $\ell \geq 2\log n$ has weight at most $O(\epsilon^{-1}\text{diam}(S_n)/n^2)$; and the total weight of these edges is $O(\epsilon^{-1}\text{diam}(S_n)) \leq O(\epsilon^{-1}\text{OPT})$. It remains to bound the weight of the edges on levels $\ell = 1, \ldots, [2\log n]$. We consider each level separately.

**Short edges.** For every edge $ab \in E$, let $B_{ab}$ denote the ellipsoid with foci $a$ and $b$, and great axis of length $(1+\epsilon)||ab||$. Note that every $ab$-path of weight at most $(1+\epsilon)||ab||$ lies in $B_{ab}$. The set of directions of line segments in $\mathbb{R}^d$ is represented by a hemisphere of $\mathbb{S}^{d-1}$. The distance between two directions is measured by angles in the range $[0, \pi]$. Recently, Bhore and Tóth [13, Lemma 3] proved that every $ab$-path of weight at most $(1+\epsilon)||ab||$ contains edges of total weight at least $\frac{1}{2}||ab||$ that make an angle at most $\epsilon^{1/2}$ with $ab$ (i.e., they are near-parallel to $ab$); see Fig. 4(right).

Since $G^*$ is a $(1+\epsilon)$-spanner for $S_n$, it contains an $ab$-path of weight at most $(1+\epsilon)||ab||$ for every $ab \in E$. This path lies in the ellipsoid $B_{ab}$, and contains edges of $G^*$ of weight at least $\frac{1}{2}||ab||$ and with direction with at most $\epsilon^{1/2}$ from $ab$. We next define suitable disjoint sets of ellipsoids, in order to establish a lower bound on $||G^*||$. 


Edge partition by directions. First, we partition the edge set $E_t$ into subsets based on the directions of the edges. We use standard volume argument to construct a homogeneous set of directions. Let $H \subseteq S^{d-1}$ be the hemisphere of unit vectors in $\mathbb{R}^d$, then the direction vector of a line segment $ab$, denoted $\text{dir}(ab)$, is a unique point in $H$. Consider a maximal packing of $H$ with (spherical) balls of radius $\frac{1}{2}\varepsilon^{1/2}$. Since the spherical volume of $H$ is $\Theta(1)$ and the volume of each ball is $\Theta(\varepsilon^{(d-1)/2})$, the number of balls is $K = \Theta(\varepsilon^{(1-d)/2})$.

By doubling the radii of the spherical balls to $\frac{1}{4}\varepsilon^{1/2}$, we obtain a covering of $H$ with a set of balls $D = \{D_i : i = 1, \ldots, K\}$. For each spherical ball $D_i \in D$, denote by $2D_i$ the concentric ball of radius $\frac{1}{2}\varepsilon^{1/2}$. By standard packing argument, the ball $2D_i$ intersects only $O(1)$ balls in $D$ (where $d = O(1)$). We can now define a partition $E_t = \bigcup_{i=1}^{K} E_t,i$, as follows: let an $ab \in E_t$ be in $E_t,i$ if $i$ is the smallest index such that $\text{dir}(ab) \in D_i$. Now for every $i = 1, \ldots, K$, let $E^*_i$ be the set of edges $e^* \in E^*$ such that $\text{dir}(e^*) \in 2D_i$. By construction, every edge $e^* \in E^*$ lies in $O(1)$ sets $E^*_i$; consequently $\sum_{i=1}^{K} \|E^*_i\| = \Theta(\|G^*\|)$. Furthermore, for every edge $ab \in E_t,i$, all edges in $E^*$ that make an angle at most $\varepsilon^{1/2}$ with $ab$ are in $E^*_i$.

Disjoint ellipsoids. For every $i = 1, \ldots, K$, let $B_{t,i}$ be the set of ellipsoids $B_{ab}$ with $ab \in E_t,i$. We show that $B_{t,i}$ contains a subset $B'_{t,i}$ of disjoint ellipsoids such that $|B'_{t,i}| \geq \Omega(\varepsilon^{d+1}|E_{t,i}|)$.

We claim that every ellipsoid in $B_{t,i}$ intersects $O(\varepsilon^{-(d+1)})$ other ellipsoids in $B_{t,i}$. We make use of a volume argument. Let $M_t = \max\{|\|e\|| : e \in E_t\}$; and note that the side length of every cube at level $\ell$ of the quadtree is $\Theta(\varepsilon M_t)$.

For every ellipsoid $B_{ab} \in B_{t,i}$, the great axis has length $(1 + \varepsilon)\|ab\|$, and the $d - 1$ minor axes each have length $\sqrt{(1 + \varepsilon)^2 - 1^2}\|ab\| < 2\varepsilon^{1/2}\|ab\|$, where $\|ab\| \leq M_t$. Hence $B_{ab}$ is contained in a cylinder $C_{ab}$ of height $(1 + \varepsilon)M_t$ whose base is a $(d - 1)$-dimensional ball of diameter $2\varepsilon^{1/2}M_t$. Any other ellipsoid in $B_{t,i}$, with great axis parallel to $ab$ is contained in a translate of $C_{ab}$. If we rotate $B_{ab}$ about its center by an angle at most $\varepsilon^{1/2}$, then its orthogonal projection to the original great axis decreases, and the maximum distance from the original great axis increases by at most $\|ab\|\frac{\varepsilon^{1/2}}{2} \sin \varepsilon^{1/2} < M_t\varepsilon^{1/2}$. Consequently, every ellipsoid in $B_{t,i}$ is contained in a translated copy of $2C_{ab}$. Hence, every ellipsoid in $B_{t,i}$ that intersects $B_{ab}$ is contained in $3C_{ab}$. Every cube at level $\ell$ of the quadtree that intersects $3C_{ab}$ is contained in the Minkowski sum of $3C_{ab}$ and such a cube, which is in turn contained in $4C_{ab}$. Note that the volume of the cylinder $4C_{ab}$ is $O(\varepsilon^{d-1/2}M_t^d)$; while the volume of a cube at level $\ell$ of the quadtree is $\Theta(\varepsilon^d M_t^d)$. Therefore $4C_{ab}$ contains $O(\varepsilon^{(d-1)/2}/\varepsilon^d) = O(\varepsilon^{-(d+1)/2})$ such cubes. Recall that the algorithm maintains one representative from each cube, and the edges $ab \in E_{t,i}$ are pairs of representative. Thus $O(\varepsilon^{-(d+1)/2})$ representatives in $4C_{ab}$ can form $O(\varepsilon^{-(d+1)})$ pairs (i.e., edges, hence ellipsoids).

This completes the proof of the claim that every ellipsoid in $B_{t,i}$ intersects $O(\varepsilon^{-(d+1)})$ other ellipsoids in $B_{t,i}$. Hence the intersection graph of $B_{t,i}$ is $O(\varepsilon^{-(d+1)})$-degenerate; and has an independent set $B'_{t,i}$ of size $|B'_{t,i}| \geq \Omega(\varepsilon^{d+1}|E_{t,i}|) = \Omega(\varepsilon^{d+1}|E_{t,i}|)$.

Weight analysis. As noted above, all edges in $E_t$ have length $\Theta(M_t)$. For every $i = 1, \ldots, K$ and for every ellipsoid $B_{ab} \in B_{t,i}$, we have $\|E^*_i \cap B_{ab}\| \geq \frac{1}{2}\|ab\|\Omega(M_t)$. Summing over a set of disjoint ellipsoids, we obtain

$$\|E^*_i\| \geq \sum_{B_{ab} \in B'_{t,i}} \|E^*_i \cap B_{ab}\| \geq \sum_{B_{ab} \in B'_{t,i}} \frac{1}{2}\|ab\|$$

$$\geq |B'_{t,i}| \cdot \frac{1}{2} \min\{\|ab\| : ab \in E_{t,i}\}$$

$$\geq \varepsilon^{-(d+1)}|E_{t,i}| \cdot \Omega(M_t) = \Omega(\varepsilon^{-(d+1)}|E_{t,i}|).$$
Summation over all directions \( i = 1, \ldots, K \) yields
\[
\|G^*\| = \Theta \left( \sum_{i=1}^{K} \|E^*_i\| \right) \geq \Omega \left( \sum_{i=1}^{K} \varepsilon^{-(d+1)} \|E_{\ell,i}\| \right) = \Omega(\varepsilon^{-(d+1)} \|E_\ell\|).
\]
Finally, summation over all \( \ell \geq 1 \) yields
\[
\|E\| = \sum_{\ell \geq 1} \|E_\ell\| \leq \sum_{\ell=1}^{[2 \log n]} \|E_\ell\| + \sum_{\ell > [2 \log n]} \|E_\ell\| \leq \varepsilon^{-(d+1)} \|G^*\| \log n + \varepsilon^{-1} \|G^*\|,
\]
as required.

### 3.2 Online Algorithm with Steiner Points

When Steiner points are allowed, we can substantially improve the competitive ratio in terms of \( \varepsilon \). We describe an algorithm with competitive ratio \( O(\varepsilon^{1-d/2} \log n) \). As a counterpart, we show in Section 4 that the dependence on \( n \) is unavoidable in dimensions \( d \geq 2 \); it remains an open problem whether the dependence on \( \varepsilon \) is necessary.

**Theorem 5.** For every \( \varepsilon > 0 \), an online algorithm can maintain, for a sequence of \( n \in \mathbb{N} \) points in the plane, a Euclidean Steiner \( (1 + \varepsilon) \)-spanner of weight \( O(\varepsilon^{-1/2} \log n) \cdot \text{OPT} \).

**Proof.** Our online algorithm has two stages: \( A_1 \) and \( A_2 \). Algorithm \( A_1 \) is the same as in Section 3.1, it maintains a quadtree \( T_n \) for the point set \( S_n \), and a “primary” \( (1 + \varepsilon) \)-spanner \( G_1 \) without Steiner points. Algorithm \( A_2 \) maintains a Steiner \( (1 + 3\varepsilon) \)-spanner \( G_2 \) as follows: for each edge \( ab \) in \( G_1 \), it creates an \( ab \)-path of length \( (1 + \varepsilon)\|ab\| \) using Steiner points in \( G_2 \). Importantly, algorithm \( A_2 \) can bundle together “similar” edges of \( G_1 \), and handle them together using shallow-light trees [52].

In particular, we partition the space of all possible edges of \( G_1 \) into buckets (edges with similar directions, locations, and weights). For each bucket \( U \), when algorithm \( A_1 \) inserts the first edge \( ab \in U \) into \( G_1 \), then algorithm \( A_2 \) creates a “backbone” Steiner tree \( T = T(U) \) of weight \( O(\|ab\|) \), which contains an \( ab \)-path of length at most \( (1 + \varepsilon)\|ab\| \). For any subsequent edge \( a'b' \in U \), it suffices to add paths from \( a' \) and \( b' \) to \( T \), of weight \( O(\varepsilon \|ab\|) \), to obtain \( a'b' \)-path of length at most \( (1 + \varepsilon)\|a'b'\| \). Overall, between any two points \( s_i, s_j \in S \), the primary spanner contains a path of weight at most \( (1 + \varepsilon)\|s_i s_j\| \), and \( G_2 \) contains an Steiner path of weight at most \( (1 + 3\varepsilon)^2 \|s_i s_j\| < (1 + 3\varepsilon)\|s_i s_j\| \), as claimed.

It remains to define the buckets \( U \), the backbone \( T(U) \) for the first edge in \( U \), and the “connectors” added for each subsequent edge in \( U \). We first describe the algorithm in the plane, where we establish a competitive ratio \( O(\varepsilon^{-1/2} \log n) \), and then generalize the construction to higher dimensions.

**Buckets.** We define buckets for all potential edges in the primary spanner \( G_1 \). We analyze a single level \( \ell \) of the quadtree \( T \). Without loss of generality, assume that the side length of all quadtree cubes in level \( \ell \) have unit length, hence the weight of every edge in \( E_\ell \) is \( \Theta(\varepsilon^{-1}) \).

In Section 3.2, we have covered the set \( H \subset S^1 \) of directions with a set \( D = \{D_i : i = 1, \ldots, K\} \) of balls of diameter \( \varepsilon^{1/2} \). For each ball in \( D \), we define a set of buckets. Let \( D \in D \), and let \( L \) be a line such that \( \text{dir}(L) \) corresponds to the center of \( D \); refer to Fig. 5(left). Partition the plane into parallel strips of width \( \frac{1}{2} \varepsilon^{1/2} \) by a set of lines parallel to \( L \); and partition each strip further into rectangles of height \( 2\varepsilon^{-3} \). By scaling up the rectangles by a factor of 2, we obtain a covering of the square \( Q_0 \) with a set \( \mathcal{R} \) of \( 4\varepsilon^{-4} \times \varepsilon^{1/2} \) rectangles such that each point is covered by \( O(1) \) rectangles in \( \mathcal{R} \).
For each rectangle $R \in \mathcal{R}$, we create a bucket $U$ comprising all edges $ab \in E_\ell$ such that $ab \subseteq R$ and $\text{dir}(ab) \in D$ (hence $\angle(\text{dir}(ab), \text{dir}(L)) \leq \varepsilon^{1/2}$). Note that every edge $ab \in E_\ell$ lies in at least one and at most $O(1)$ buckets.

**Figure 5** Left: The overlay the the quadtree with a partition of $\mathbb{R}^2$ into $\frac{1}{2} \varepsilon^{-1/2} \times 2\varepsilon^{-1}$ rectangles aligned with $L$. Top-Right: A rectangle $R \in \mathcal{R}$, the grid $G(U)$, and the partition of $R$ into $\varepsilon^{-1/2} \times \varepsilon^{-1/2}$ squares. Bottom-Right: A shallow-light tree between a side of an $\frac{1}{2} \varepsilon^{-1/2} \times \frac{1}{2} \varepsilon^{-1/2}$ square and a source $r \in L(U)$.

**Backbones and Connectors.** Let $U$ be a bucket defined above for a rectangle $R \in \mathcal{R}$. Let $L(U)$ denote the median of the rectangle $R$ parallel to $L$. When the primary algorithm $A_1$ inserts the first edge $ab \in U$ into $G_1$, then Algorithm $A_2$ constructs a unit grid graph $G(U)$, formed by a subdivision of $R$ into unit squares; see Fig. 5(top-right). Since $R$ is a $4\varepsilon^{-1} \times \varepsilon^{-1/2}$ rectangle, $\|G(U)\| = O(\varepsilon^{-3/2})$. Furthermore, we partition $R$ into $4\varepsilon^{-1/2}$ squares of side length $\varepsilon^{-1/2}$. For each such square, we insert two shallow-light trees [52] between the two sides of the square orthogonal to $L$ and two points in $L(U)$ at distance $\varepsilon^{-1}$ from the square on either side; Fig. 5(bottom-right). The weight of each shallow-light tree is $O(\varepsilon^{-1})$ [52], and so the combined weight of $O(\varepsilon^{-1/2})$ shallow-light trees is $O(\varepsilon^{-3/2})$. The grid $G(U)$ together with the shallow-light trees forms the **backbone** for the bucket $U$ in $G_2$.

We add **connector** edges between $a$ (resp., $b$) and the four corners of unit square of the grid $G(U)$ that contains it. For any subsequent edge $a'b' \in U$ that algorithm $A_1$ inserts into $G_1$, the backbone does not change, we only add connectors between $a'$ (resp., $b'$) and the four corners of the unit square in $G(U)$ that contains it. The weight of the four connectors is $O(1)$ per point. Since area($R) = \Theta(\varepsilon^{-3/2})$, then $R$ intersects at most $O(\varepsilon^{-3/2})$ unit squares of the quadtree at level $\ell$, and so the total weight of all connectors is $O(\varepsilon^{-3/2})$, as well.

**Stretch analysis.** Suppose algorithm $A_1$ inserts an edge $cd$ into $G_1$. As noted above, $cd$ lies in $\Theta(1)$ buckets; refer to Fig. 6. Suppose bucket $U$ contains $cd$; and in the partition of the rectangle $R = R(U)$, the endpoint $c$ ($d$) lies squares $R_c$ ($R_d$) of side length $\varepsilon^{-1/2}$, associated with shallow-light trees rooted at $r_c$ ($r_d$). Then $G_2$ contains a $cd$-path comprised of: (i) connectors from $c$ and $d$, resp., to the closest point in the grid $G(U)$; (ii) paths in $G(U)$ from the connectors to the boundary of squares $R_c$ and $R_d$, (iii) paths along the shallow-light trees to the roots $r_c, r_d \in L(U)$, and (iv) the line segment $r_c r_d$ in $G(U)$. The weight of each connector in (i) is at most $2\sqrt{2}$, which is bounded by $O(\varepsilon)\|cd\|$ since $|cd| = \Theta(\varepsilon^{-1})$. The edges in (ii) and (iv) are parallel to $L$, hence they make an angle less than $\varepsilon^{1/2}$ with
Finally, consider the two subpaths in part (iii) in shallow-light trees: The line segment between the two endpoints of each such subpath makes an angle less than \(\varepsilon^{1/2}\) with \(L\), hence less than \(2\varepsilon^{1/2}\) with \(cd\); and the weight of a root-to-leaf path in a shallow-light tree is a \((1 + \varepsilon)\)-approximation of the straight-line segment between its endpoints. Overall, the total weight of the \(cd\)-path described above is \((1 + O(\varepsilon))\|cd\|\), as required.

For every point pair \(a, b \in S_n\), the primary graph \(G_1\) contains an \(ab\)-path \(P = (p_0, \ldots, p_m)\) of length \(\|P\| \leq (1 + \varepsilon)\|ab\|\), since \(G_1\) is a \((1 + \varepsilon)\)-spanner. We have shown that for every edge \(p_{i-1}p_i\) of \(G_1\), the Steiner spanner \(G_2\) contains a \(p_{i-1}p_i\)-path of weight \((1 + O(\varepsilon))\|p_{i-1}p_i\|\). The concatenation of these paths yields an \(ab\)-path in \(G_2\), of weight \(\sum_{i=1}^m (1 + O(\varepsilon))\|p_{i-1}p_i\| = (1 + O(\varepsilon))\|P\| = (1 + O(\varepsilon))(1 + \varepsilon)\|ab\| = (1 + O(\varepsilon))\|ab\|\).

**Competitive Analysis.** Denote by \(E_\ell\) the set of edges of \(G_2\) added at level \(\ell = 1, \ldots, 2\log n\), and let \(b_\ell\) be the number of nonempty buckets at level \(\ell\). We have seen that for each nonempty bucket at level \(\ell\), \(E_\ell\) contains a subgraph of weight \(O(\varepsilon^{-3/2}\text{diam}(S_n)/2^\ell)\); hence \(\|E_\ell\| \leq O(b_\ell \cdot \varepsilon^{-3/2}\text{diam}(S_n)/2^\ell)\).

Let \(G^* = (S_n, E^*)\) the a Euclidean Steiner \((1 + \varepsilon)\)-spanner for \(S_n\) of minimum weight OPT. Consider a nonempty bucket \(U\) associated with a line \(L\) and a rectangle \(R(U)\). Since \(U\) is nonempty, there is an edge \(ab \in U\) in \(G_1\). Recall that \(ab \in R\) and \(\angle(\text{dir}(ab), \text{dir}(L)) \leq \varepsilon^{1/2}\). Since \(G^*\) is a \((1 + \varepsilon)\)-spanner, it contains an \(ab\)-path \(P_{ab}\) of weight at most \((1 + \varepsilon)\|ab\|\).

As noted in Section 3.1, \(P_{ab}\) lies in the ellipse \(B_{ab}\), and contains edges of weight at least \(\frac{1}{2}\|ab\|\) that make an angle at most \(\varepsilon^{1/2}\) with \(ab\). All points in the ellipse \(B_{ab}\) are at distance less than \(\varepsilon^{1/2}\) from the the line segment \(ab\). The segment \(ab\) lies in the \(4\varepsilon^{-1} \times \varepsilon^{1/2}\) rectangle \(R(U)\). Thus we have \(P_{ab} \subseteq B_{ab} \subseteq 2R(U)\), and so \(2R(U)\) contains edges of \(G^*\) of weight \(\frac{1}{2}\|ab\| = O(\varepsilon^{-1}\text{diam}(S_n)/2^\ell)\) whose directions are within \(2\varepsilon^{1/2}\) from \(L\); denote by \(E^*(U) \subseteq E^*\) the set of these edges. By construction, each edge \(e^*\) of \(G^*\) lies in \(E^*(U)\) for only \(O(1)\) buckets. Indeed, there are \(O(1)\) lines \(L'\) with \(\angle(\text{dir}(L), \text{dir}(L')) \leq 2\varepsilon^{1/2}\), and for each such direction \(L'\), every point in \(\mathbb{R}^2\) lies in \(O(1)\) rectangles \(2R(U')\) aligned with \(L'\). We conclude that \(\text{OPT} = \|G^*\| = \Omega(b_\ell \cdot \varepsilon^{-1}\text{diam}(S_n)/2^\ell)\). This implies \(\|E_\ell\|/\text{OPT} \leq O(\varepsilon^{-1/2})\) for \(\ell = 1, \ldots, 2\log n\). Summation over all levels yields

\[
\frac{\text{ALG}}{\text{OPT}} = \sum_{\ell=1}^{2\log n} \frac{\|E_\ell\|}{\text{OPT}} \leq \sum_{\ell=1}^{2\log n} O(\varepsilon^{-1/2}) + O(1) = O(\varepsilon^{-1/2}\log n),
\]

as claimed.

**Generalization to \(\mathbb{R}^d\).** Our algorithm and its analysis generalize to Euclidean \(d\)-space.

**Theorem 6.** For every \(\varepsilon > 0\), an online algorithm can maintain, for a sequence of \(n \in \mathbb{N}\) points in \(\mathbb{R}^d\), a Euclidean Steiner \((1 + \varepsilon)\)-spanner of weight \(O(\varepsilon^{(1-d)/2}\log n) \cdot \text{OPT}\).

The proof is analogous to that of Theorem 5. The bottleneck of the competitive analysis is the size of the unit grids \(G(U)\) which is \(\Theta(\varepsilon^{-(d+1)/2})\) in \(\mathbb{R}^d\), and it is contrasted with a path of weight \(\Omega(\varepsilon^{-1})\) in \(\text{OPT}\). Similarly to Section 3.1, we choose a homogeneous set \(D\) of
\[ \Theta(\varepsilon^{(1-d)/2}) \] directions. For each direction \( L \in D \), we construct a tiling of \( \mathbb{R}^d \) with congruent hyper-rectangles aligned with \( L \) of dimensions \( \varepsilon^{-1} \times \varepsilon^{-1/2} \times \ldots \times \varepsilon^{-1/2} \). Refer to the full paper for all further details.

4 Lower Bound with Steiner Points

Recall that when Steiner points are allowed, the algorithm may subdivide existing edges with Steiner points. It follows that the in one-dimension, an online algorithm can maintain a Hamiltonian path on \( S_n \), which is the minimum \((1 + \varepsilon)\)-spanner for all \( \varepsilon \geq 0 \). This property carries over to Euclidean Steiner 1-spanners (i.e., the case \( \varepsilon = 0 \)), where we need to maintain the complete straight-line graph on \( n \) points. However, we show that for \( \varepsilon > 0 \) in dimensions \( d \geq 2 \), the competitive ratio of an online algorithm with Steiner points must depend on \( n \).

\[ \text{Theorem 7.} \quad \text{For every} \ \varepsilon > 0, \ \text{the competitive ratio of any online algorithm that maintains a Euclidean Steiner} \ (1 + \varepsilon)-\text{spanner for a sequence of} \ n \ \text{points in} \ \mathbb{R}^d \ \text{is} \ \Omega(f(n)) \ \text{for some function} \ f(n) \ \text{such that} \ \lim_{n \to \infty} f(n) = \infty. \]

\[ \text{Proof.} \quad \text{We describe and analyze an adversarial strategy for placing points in the plane in stages. In stage 1, the adversary places two points at} \ s = (0,0) \ \text{and} \ t = (1,0), \ \text{both on the} \ x\text{-axis. In subsequent stages, new points are arranged so that the optimum solution remains an} \ x\text{-monotone path of length at most} \ 1 + \varepsilon \ \text{at all times.} \]

\[ \text{Let us denote by} \ A_i \ \text{the points placed in stage} \ i. \ \text{At the end of stage} \ i, \ \text{adversary constructs the point set} \ A_{i+1} \ \text{based on the current} \ (1 + \varepsilon)-\text{spanner built by the algorithm} \ \text{ALG, and then placed the points in} \ A_{i+1} \ \text{in an arbitrary order. The objective is that in each stage,} \ \text{ALG has to add new edges of total weight at least} \ 1/2. \ \text{Since} \ \text{OPT} \leq 1 + \varepsilon \ \text{at all times, and} \ \text{ALG} \geq \frac{1}{2}(i+1) \ \text{after} \ i \ \text{stages, the competitive ratio goes to infinity.} \]

\[ \text{We describe stage 2 in more detail; subsequent stages are similar; see Fig. 7. At the end of stage 1, our point set is} \ A_1 = \{s = (0,0), t = (1,0)\}, \ \text{the optimal spanner is a single edge of unit weight, and} \ \text{ALG has constructed a Euclidean Steiner} \ (1 + \varepsilon)-\text{spanner} \ G_1 \ \text{for} \ A_1. \ \text{Let} \ k_1 = \left[ \|G_1\| \right]. \ \text{The adversary considers} \ 2k_1 + 1 \ \text{circular arcs between} \ s \ \text{ad} \ t, \ \text{each of weight at most} \ 1 + \frac{\varepsilon}{2}. \ \text{Let} \ \text{ALG constructs the point set} \ A_{i+1} \ \text{based on the current} \ (1 + \varepsilon)-\text{spanner built by the algorithm} \ \text{ALG, and then placed the points in} \ A_{i+1} \ \text{in an arbitrary order. The objective is that in each stage,} \ \text{ALG has to add new edges of total weight at least} \ 1/2. \ \text{Since} \ \text{OPT} \leq 1 + \varepsilon \ \text{at all times, and} \ \text{ALG} \geq \frac{1}{2}(i+1) \ \text{after} \ i \ \text{stages, the competitive ratio goes to infinity.} \]

\[ \text{We describe stage 2 in more detail; subsequent stages are similar; see Fig. 7. At the end of stage 1, our point set is} \ A_1 = \{s = (0,0), t = (1,0)\}, \ \text{the optimal spanner is a single edge of unit weight, and} \ \text{ALG has constructed a Euclidean Steiner} \ (1 + \varepsilon)-\text{spanner} \ G_1 \ \text{for} \ A_1. \ \text{Let} \ k_1 = \left[ \|G_1\| \right]. \ \text{The adversary considers} \ 2k_1 + 1 \ \text{circular arcs between} \ s \ \text{ad} \ t, \ \text{each of weight at most} \ 1 + \frac{\varepsilon}{2}. \ \text{The arcs define} \ 2k_1 \ \text{interior-disjoint bounded regions. Let} \ R_1 \ \text{be a region that minimizes the weight} \ \|G_1 \cap R_1\|, \ \text{in particular,} \ \|G_1 \cap R_1\| \leq \frac{1}{2k_1} \|G_1\| \leq \frac{1}{2}. \ \text{In the interior of} \ R_1, \ \text{let} \ \gamma_1 \ \text{be another circular arc between} \ s \ \text{and} \ t, \ \text{of weight} \ \|\gamma_1\| \leq 1 + \frac{\varepsilon}{2}; \ \text{and let} \ A_2 = \{t_1, \ldots, t_N\} \ \text{be a set of points along} \ \gamma_1, \ \text{labeled in} \ x\text{-monotone increasing order with the following properties: (1) For every} \ i = 1, \ldots, N - 1, \ \text{the ellipse} \ B_i \ \text{with foci} \ t_i \ \text{and} \ t_{i+1}, \ \text{and great axis} \ (1 + \varepsilon)\|t_it_{i+1}\| \ \text{lies entirely in} \ R_1; \ \text{and (2) the weight of the} \ x\text{-monotone path} \ \{t_1, t_2, \ldots, t_N\} \ \text{is at least} \ 1. \]

\[ \text{Figure 7 Left:} \ \text{For} \ A_1 = \{s, t\}, \ \text{ALG constructs a} \ (1+\varepsilon)-\text{spanner} \ G_1 \ \text{(red).} \ \text{Five circular arcs define four regions; region} \ R_1 \ \text{satisfies} \ \|G_1 \cap R_1\| \leq \frac{1}{2}\|G_1\|. \ \text{In stage 2, the adversary presents points} \ A_2 \ \text{in} \ R_1; \ \text{Middle:} \ \text{The algorithm augments} \ G_1 \ \text{to} \ G_2. \ \text{Right:} \ \text{Region} \ R_2 \ \text{satisfies} \ \|G_2 \cap R_2\| \leq \frac{1}{2k_2} \|G_2\|. \]

\[ \text{In stage 2, the adversary presents the points in} \ A_2 \ \text{in an arbitrary order. By the end of stage 2,} \ \text{ALG augments} \ G_1 \ \text{to a Euclidean Steiner} \ (1 + \varepsilon)-\text{spanner} \ G_2 \ \text{for} \ A_1 \cup A_2. \ \text{In particular, for every} \ i = 1, \ldots, N - 1, \ \text{the graph} \ G_2 \ \text{contains a} \ t_it_{i+1}-\text{path of length at most} \ (1 + \varepsilon)\|t_it_{i+1}\|, \ \text{which lies in the ellipse} \ E_i, \ \text{here in the interior of the region} \ R_1. \ \text{The part of} \]

\[ \text{Figure 7 Left:} \ \text{For} \ A_1 = \{s, t\}, \ \text{ALG constructs a} \ (1+\varepsilon)-\text{spanner} \ G_1 \ \text{(red).} \ \text{Five circular arcs define four regions; region} \ R_1 \ \text{satisfies} \ \|G_1 \cap R_1\| \leq \frac{1}{2}\|G_1\|. \ \text{In stage 2, the adversary presents points} \ A_2 \ \text{in} \ R_1; \ \text{Middle:} \ \text{The algorithm augments} \ G_1 \ \text{to} \ G_2. \ \text{Right:} \ \text{Region} \ R_2 \ \text{satisfies} \ \|G_2 \cap R_2\| \leq \frac{1}{2k_2} \|G_2\|. \]

\[ \text{In stage 2, the adversary presents the points in} \ A_2 \ \text{in an arbitrary order. By the end of stage 2,} \ \text{ALG augments} \ G_1 \ \text{to a Euclidean Steiner} \ (1 + \varepsilon)-\text{spanner} \ G_2 \ \text{for} \ A_1 \cup A_2. \ \text{In particular, for every} \ i = 1, \ldots, N - 1, \ \text{the graph} \ G_2 \ \text{contains a} \ t_it_{i+1}-\text{path of length at most} \ (1 + \varepsilon)\|t_it_{i+1}\|, \ \text{which lies in the ellipse} \ E_i, \ \text{here in the interior of the region} \ R_1. \ \text{The part of} \]
the path between the vertical lines passing through \( t_i \) and \( t_{i+1} \) has weight at least \( \|t_i t_{i+1}\| \).
Since these parts are disjoint, the total weight all \( N - 1 \) paths is at least \( \sum_{i=1}^{N-1} \|t_i t_{i+1}\| \geq 1 \).
Consequently, \( \|G_2 \cap R_i\| \geq 1 \). Since we had \( \|G_1 \cap R_i\| \leq \frac{1}{2} \), ALG must have added new edges of weight at least \( \frac{1}{2} \) in stage 2, as claimed.

In phase \( i + 1 \), in general, let \( k_i = \|G_i\| \). Label the points in the current point set \( S = \bigcup_{j=1}^i A_j \) by \( s_0, \ldots, s_n \) in \( x \)-monotone order, and assume that the \( x \)-monotone path spanned by \( S \) has weight \( \text{OPT} = 1 + (1 - \frac{1}{2})\varepsilon \).
For all segments \( s_j s_{j+1} \), we consider \( 2k_i + 1 \) \( x \)-monotone circular arcs such that the total weight of any concatenation of the circular arcs from \( s = s_0 \) to \( t = s_n \) is at most \( 1 + (1 - \frac{1}{2\sqrt{d}})\varepsilon \).
For each segment \( s_j s_{j+1} \), we choose one of \( 2k_i \) regions that has a minimum-weight intersection with \( G_i \), and let \( R_i \) be the union of these regions. Note that \( \|G_i \cap R_i\| \leq \frac{1}{2k_i} \|G_i\| \leq \frac{1}{2} \).
Let \( \gamma_i \) be an \( st \)-path \( \gamma_i \) that connects the points \( s_0, \ldots, s_n \) via circular arcs in the region \( R_i \), and has weight at most \( 1 + (1 - \frac{1}{2\sqrt{d}})\varepsilon \).
Now the adversary can choose a finite point set \( A_{i+1} = \{t_1, \ldots, t_N\} \) along \( \gamma_i \) with properties (1)–(2) above. This completes the description of the adversarial strategy.

Similarly to stage 2, when ALG augments \( G_i \) to a Euclidean Steiner \((1 + \varepsilon)\)-spanner \( G_{i+1} \) for \( \bigcup_{j=1}^i A_j \), he must add new edges of weight at least \( \frac{1}{2} \) in the region \( R_i \). It follows that the competitive ratio for any online algorithm goes to infinity as \( n \) goes to infinity.

\section{Conclusions}

We have studied online spanners for sequences of points in \( \mathbb{R}^d \), in fixed dimensions \( d \geq 1 \), under \( L_2 \) and \( L_1 \) norms. We established a tight bound of \( \Theta(\varepsilon^{-1} \log n / \log \varepsilon^{-1}) \) for the competitive ratio of any online \((1 + \varepsilon)\)-spanner algorithms on a real line (Theorem 3). However, it remains an open problem to close the gap between the lower and upper bounds in \( \mathbb{R}^d \), for \( d \geq 2 \).
Under the \( L_2 \) norm, previously known algorithms achieve competitive ratio \( O(\varepsilon^{-(d+1)} \log n) \) (Theorem 4). The best lower bound we are aware of holds for \( d = 1 \). It is unclear whether the lower bound can be improved to \( \varepsilon^{-\omega(d)} \log n \) for \( d \geq 2 \).

Next, we have shown that, if an online algorithm is allowed to use Steiner points, it can achieve a substantially better competitive ratio in terms of \( \varepsilon \), namely \( O(\varepsilon^{(1-d)/2} \log n) \), for a sequence of \( n \) points in \( \mathbb{R}^d \) and any constant \( d \geq 2 \), under the \( L_2 \) norm (Theorem 6). As a counterpart, we proved that any online spanner algorithm for a sequence of \( n \) points in \( \mathbb{R}^d \) under \( L_2 \) norm has competitive ratio \( \Omega(f(n)) \), where \( \lim_{n \to \infty} f(n) = \infty \) (Theorem 7). It remains an open problem whether the competitive ratio depends on \( \varepsilon \) for Euclidean Steiner spanners. Another open problem is whether the factor \( \log n \) in the upper bounds can be reduced, e.g., to \( \log n / \log \log n \); similar to the work by Alon and Azar [2] who established such a lower bound for Euclidean minimum Steiner trees (EMST) for \( n \) points in \( \mathbb{R}^2 \).

We have established a lower bound \( \Omega(\varepsilon^{-d}) \) for the competitive ratio under the \( L_1 \)-norm in \( \mathbb{R}^d \). It is unclear whether it can be improved by a \( \log n \) factor in dimensions \( d \geq 2 \). Designing online algorithms that match these bounds under the \( L_1 \) norm is left for future research.

In online spanner algorithms, the decisions are irrevocable, which means that once an edge is added to the spanner by an online algorithm, it can never be deleted. However, if some of the decisions are reversible, better bounds may be possible. This model is commonly known as \textit{online algorithms with recourse} [33, 39, 45]. In 1-dimensional, for instance, an optimum spanner is just a monotone path connecting the points in linear order, and any online algorithm that is allowed to remove at least one edge at per iteration can maintain such a path. In higher dimensions, however, it is unclear whether a \( O(1) \)-approximation of the minimum-weight \((1 + \varepsilon)\)-spanner can be maintained with \( O(\varepsilon^{-d+1}) \) recourse.
## References


16:18 Online Euclidean Spanners

22 L. Paul Chew. There is a planar graph almost as good as the complete graph. In *Proc. 2nd Symposium on Computational Geometry (SoCG)*, pages 169–177. ACM Press, 1986.


