Online Spanners in Metric Spaces

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

Arnold Filtser
Department of Computer Science, Bar-Ilan University, Ramat-Gan Israel

Hadi Khodabandeh
Department of Computer Science, University of California, Irvine, CA, USA

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

Abstract

Given a metric space $\mathcal{M} = (X, \delta)$, a weighted graph G over X is a metric t-spanner of $\mathcal{M}$ if for every $u, v \in X$, $\delta(u,v) \le \delta_G(u,v) \le t \cdot \delta(u,v)$, where δ_G is the shortest path metric in G. In this paper, we construct spanners for finite sets in metric spaces in the online setting. Here, we are given a sequence of points $(s_1, \ldots, s_n)$, where the points are presented one at a time (i.e., after i steps, we have seen $S_i = \{s_1, \ldots, s_i\}$). The algorithm is allowed to add edges to the spanner when a new point arrives, however, it is not allowed to remove any edge from the spanner. The goal is to maintain a t-spanner G_i for S_i for all i, while minimizing the number of edges, and their total weight.

Under the L_2-norm in $\mathbb{R}^d$ for arbitrary constant $d \in \mathbb{N}$, we present an online $(1+\varepsilon)$-spanner algorithm with competitive ratio $O_d(\varepsilon^{-d} \log n)$, improving the previous bound of $O_d(\varepsilon^{-(d+1)} \log n)$. Moreover, the spanner maintained by the algorithm has $O_d(\varepsilon^{1-d} \log \varepsilon^{-1}) \cdot n$ edges, almost matching the (offline) optimal bound of $O_d(\varepsilon^{1-d}) \cdot n$. In the plane, a tighter analysis of the same algorithm provides an almost quadratic improvement of the competitive ratio to $O(\varepsilon^{-3/2} \log \varepsilon^{-1} \log n)$, by comparing the online spanner with an instance-optimal spanner directly, bypassing the comparison to an MST (i.e., lightness). As a counterpart, we design a sequence of points that yields a $\Omega_d(\varepsilon^{-d})$ lower bound for the competitive ratio for online $(1+\varepsilon)$-spanner algorithms in $\mathbb{R}^d$ under the L_1-norm.

Then we turn our attention to online spanners in general metrics. Note that, it is not possible to obtain a spanner with stretch less than 3 with a subquadratic number of edges, even in the offline setting, for general metrics. We analyze an online version of the celebrated greedy spanner algorithm, dubbed *ordered greedy*. With stretch factor $t = (2k-1)(1+\varepsilon)$ for $k \ge 2$ and $\varepsilon \in (0,1)$, we show that it maintains a spanner with $O(\varepsilon^{-1} \log \varepsilon^{-1}) \cdot n^{1+\frac{1}{k}}$ edges and $O(\varepsilon^{-1} n^{\frac{1}{k}} \log^2 n)$ lightness for a sequence of n points in a metric space. We show that these bounds cannot be significantly improved, by introducing an instance that achieves an $\Omega(\frac{1}{k} \cdot n^{1/k})$ competitive ratio on both sparsity and lightness. Furthermore, we establish the trade-off among stretch, number of edges and lightness for points in ultrametrics, showing that one can maintain a $(2+\varepsilon)$-spanner for ultrametrics with $O(\varepsilon^{-1} \log \varepsilon^{-1}) \cdot n$ edges and $O(\varepsilon^{-2})$ lightness.

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

Keywords and phrases spanner, online algorithm, lightness, sparsity, minimum weight

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.18

Related Version *Full Version*: https://arxiv.org/abs/2202.09991

Funding *Csaba D. Tóth*: NSF DMS-1800734.

30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 18; pp. 18:1–18:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Let $\mathcal{M} = (P, \delta)$ be a finite metric space. Let $G = (P, E)$ be a graph on the points of P in $\mathcal{M}$, where the edges are weighted with the distances between their endpoints. The graph G is a *t-spanner*, for $t \geq 1$, if $\delta_G(u, v) \leq t \cdot \delta(u, v)$ for all $u, v \in P$, where $\delta_G(u, v)$ is the length of the shortest path between u and v in G, and $\delta(u, v)$ is the distance between u and v in $\mathcal{M}$.[1] The *stretch factor* t of G is the maximum distortion between the metrics δ and δ_G. Spanners were first introduced by Peleg and Schäffer [52], and since then they have turned out to be one of the fundamental graph structures with numerous applications in the area of distributed systems and communication, distributed queuing protocol, compact routing schemes, etc. [25, 43, 53, 54].

The study of Euclidean spanners, where $P \subset \mathbb{R}^d$ with L_2-norm, was initiated by Chew [23]. Since then a large body of research has been devoted to Euclidean spanners due to its vast range of applications across domains, such as topology control in wireless networks, efficient regression in metric spaces, approximate distance oracles, data structures, and many more [34, 38, 57, 60]. Some of the results generalize to metric spaces with constant doubling dimensions [18] (the doubling dimension of $\mathbb{R}^d$ under L_2-norm is $\Theta(d)$).

Lightness and *sparsity* are two fundamental parameters for spanners. The lightness of a spanner $G = (P, E)$ is the ratio $w(G)/w(MST)$ between the total weight of G and the weight of a minimum spanning tree (MST) on P. The sparsity of G is the ratio $|E(G)|/|E(MST)| \approx |E(G)|/|P|$ between the number of edges of G and an MST. Since every spanner is connected and thus contain a spanning tree, the lightness and sparsity of a spanner G, resp., are trivial lower bounds for the ratio of $w(G)$ and $|E(G)|$ to the optimum weight and the number of edges.

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

The performance of an online algorithm ALG is measured by comparing it with the offline optimum OPT using the standard notion of competitive ratio [17, Ch. 1]. The *competitive ratio* of an algorithm ALG is defined as $\sup_\sigma \frac{\mathrm{ALG}(\sigma)}{\mathrm{OPT}(\sigma)}$, where the supremum is taken over all input sequences σ, $\mathrm{OPT}(\sigma)$ is the minimum weight of a t-spanner for the (unordered) set of points in σ, and $\mathrm{ALG}(\sigma)$ denotes the weight of the t-spanner produced by ALG for this input sequence. Note that, in order to measure the competitive ratio it is important that σ is a finite sequence of points.

The online spanner problem is motivated by natural application domains. For example, consider a developing area with limited resources, where new settlements are created successively. As the community grows, new roads are built, and there is no reason to remove existing roads. Alternatively, online spanners are also motivated by distributed mobile computing, where new subscribers successively join a network. Maintaining a cost-effective network is equivalent to minimum-weight online spanner problem.

[1] Often in the literature, the input metric is the shortest path metric of a graph $G = (V, E, w)$, and a spanner is required to be a subgraph of the input graph (see e.g. [4]). Here we study metric spanners where there is no such requirement.

In the online minimum spanning tree problem, points of a finite metric space arrive one-by-one, and we need to connect each new point to a previous point to maintain a spanning tree. Imase and Waxman [44] proved $\Theta(\log n)$-competitiveness, which is the best possible bound. Later, Alon and Azar [2] studied this problem 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 analyze the impact of auxiliary points (Steiner points) on a geometric network problem in the online setting. Several algorithms were proposed over the years for the online minimum Steiner tree and Steiner forest problems, on graphs in both weighted and unweighted settings; see [1, 5, 10, 40, 50]. However, these algorithms do not provide any guarantees on the stretch factor. This leads to the following open problem.

▶ **Problem.** *Determine the best possible bounds for the competitive ratios for the weight and the number of edges of online t-spanners, for $t \geq 1$.*

Previously, Gupta et al. [39, Theorem 1.5] constructed online spanners for terminal pairs in the same model we consider here. The analysis of [39] implicitly implies that, given a sequence of n points in an online fashion in a general metric space, one can maintain a $O(\log n)$-spanner with $O(n)$ edges and $O(\log n)$ lightness, as pointed out by one of the authors [59]. Recent work on online *directed* spanners [36] is not comparable to our results.

In the geometric setting, $(1+\varepsilon)$-spanners are possible in any constant dimension $d \in \mathbb{N}$. Tight worst-case bounds $\Theta_d(\varepsilon^{-d})$ and $\Theta_d(\varepsilon^{1-d})$ on the lightness and sparsity of offline $(1+\varepsilon)$-spanners have recently been established by Le and Solomon [47]. Online Euclidean spanners in $\mathbb{R}^d$ have been introduced by Bhore and Tóth [14]. In the real line (1D), they have established a tight bound of $O((\varepsilon^{-1}/\log\varepsilon^{-1})\log n)$ for the competitive ratio of any online $(1+\varepsilon)$-spanner algorithm for n points. In dimensions $d \geq 2$, the dynamic algorithm DefSpanner of Gao et al. [33] maintains a $(1+\varepsilon)$-spanner with $O_d(\varepsilon^{-(d+1)}n)$ edges and $O_d(\varepsilon^{-(d+1)}\log n)$ lightness, and works under the online model (as it never deletes edges when new points arrive). However, no lower bound better than the 1-dimensional $\Omega((\varepsilon^{-1}/\log\varepsilon^{-1})\log n)$ is currently known in higher dimensions.

1.1 Our Contribution

See Table 1 for an overview of our results.

Upper Bounds for Points in $\mathbb{R}^d$. Under the L_2-norm in $\mathbb{R}^d$, for arbitrary constant $d \in \mathbb{N}$, we present an online algorithm for $(1+\varepsilon)$-spanner with lightness $O_d(\varepsilon^{-d}\log n)$ and sparsity $O(\varepsilon^{1-d}\log\varepsilon^{-1})$ (Theorem 2 in Section 2.1). This improves upon the previous lightness bound of $O_d(\varepsilon^{-(d+1)}\log n)$ by Gao et al. [33, Lemma 3.8]. In the plane, we give a tighter analysis of the same algorithm and achieve an almost quadratic improvement of the *competitive ratio* to $O(\varepsilon^{-3/2}\log\varepsilon^{-1}\log n)$ (Theorem 6 in Section 2.2). Recall that in the offline setting, $\Theta(\varepsilon^{-2})$ is a tight worst-case bound for the lightness of a $(1+\varepsilon)$-spanner in the plane [47]. We obtain a better dependence on ε by comparing the online spanner with an instance-optimal spanner directly, bypassing the comparison to an MST (i.e., lightness). The logarithmic dependence on n cannot be eliminated in the online setting, based on the lower bound in $\mathbb{R}^1$ [14].

Lower Bounds for Points in $\mathbb{R}^d$. As a counterpart, we design a sequence of points that yields a $\Omega_d(\varepsilon^{-d})$ lower bound for the competitive ratio for online $(1+\varepsilon)$-spanner algorithms in $\mathbb{R}^d$ under the L_1-norm (Theorem 13 in Section 3). This improves the previous bound of $\Omega(\varepsilon^{-2}/\log\varepsilon^{-1})$ in $\mathbb{R}^2$ under the L_1-norm [14]. It remains open whether a similar lower bound holds in $\mathbb{R}^d$ under the L_2-norm; the current best lower bound is $\Omega((\varepsilon^{-1}/\log\varepsilon^{-1})\log n)$, established in [14], holds already for the real line ($d = 1$).

Table 1 Overview of online spanners algorithms. In the last three rows, we compare the spanner weight directly with the optimum weight (rather than the MST) to bound the competitive ratio.

Family	Stretch	# of edges	Lightness	Reference
General metrics	$(2k-1)(1+\varepsilon)$	$O(\varepsilon^{-1}\log\varepsilon^{-1})\, n^{1+\frac{1}{k}}$	$O(\varepsilon^{-1}n^{\frac{1}{k}}\log^2 n)$	Theorem 14
	$O(\log n)$	$O(n)$	$O(\log n)$	[39, 59]
α-HST	$2\frac{\alpha}{\alpha-1}$	$n-1$	1	Full paper
Ultrametric	$O(\varepsilon^{-1})$	$n-1$	$1+\varepsilon$	Full paper
	$2+\varepsilon$	$O(\varepsilon^{-1}\log\varepsilon^{-1})\, n$	$O(\varepsilon^{-2})$	Full paper
Doubling d-space	$1+\varepsilon$	$\varepsilon^{-O(d)}\, n$	$\varepsilon^{-O(d)}\log n$	[33]
Euclidean d-space	$1+\varepsilon$	$O_d(\varepsilon^{-d})\, n$	$O_d(\varepsilon^{-(d+1)}\log n)$	[33]
	$1+\varepsilon$	$O_d(\varepsilon^{1-d})\, n$	$\Omega(\varepsilon^{-1}n)$	[56]
	$1+\varepsilon$	$O_d(\varepsilon^{1-d}\log\varepsilon^{-1})\, n$	$O_d(\varepsilon^{-d}\log n)$	Theorem 2
Real line	$1+\varepsilon$	$O(n)$	$\tilde{\Theta}(\varepsilon^{-1}\log n)$	[14]
Family	**Stretch**	**# of edges**	**Comp. Ratio**	**Reference**
General metrics	$2k-1$	-	$\Omega(\frac{1}{k}\cdot n^{\frac{1}{k}})$	Theorem 19
Euclidean plane	$1+\varepsilon$	$O(\varepsilon^{-1}\log\varepsilon^{-1})\, n$	$\tilde{O}(\varepsilon^{-3/2}\log n)$	Theorem 6
$\mathbb{R}^d$ with L_1-norm	$1+\varepsilon$	-	$\Omega(\varepsilon^{-d})$	Theorem 13

Points in General Metrics. In Section 4, we study online spanners in general metrics. Note that it is not possible to obtain a spanner with stretch less than 3 with a subquadratic number of edges, even in the offline settings, for general metrics. We analyze an online version of the celebrated greedy spanner algorithm, dubbed *ordered greedy*. With stretch factor $t=(2k-1)(1+\varepsilon)$ for $k\geq 2$ and $\varepsilon\in(0,1)$, we show that it maintains a spanner with $O(\varepsilon^{-1}\log\varepsilon^{-1})\cdot n^{1+\frac{1}{k}}$ edges and $O(\varepsilon^{-1}n^{\frac{1}{k}}\log^2 n)$ lightness for a sequence of n points in a metric space (Theorem 14). We show (in Theorem 19) that these bounds cannot be significantly improved, by introducing an instance where every online algorithm will have $\Omega(\frac{1}{k}\cdot n^{1/k})$ competitive ratio on both sparsity and lightness. Next, we establish the trade-off among stretch, number of edges and lightness for points in ultrametrics. Specifically, we show (in the full paper) that it is possible to maintain a $(2+\varepsilon)$-spanner with $O(\varepsilon^{-1}\log\varepsilon^{-1})\cdot n$ edges and $O(\varepsilon^{-2})$ lightness in ultrametrics. Note that as the uniform metric (shortest path on a clique) is an ultrametric, any subquadratic spanner must have stretch at least 2.

1.2 Related Work

1.2.1 Dynamic & Streaming Algorithms for Graph Spanners

A t-spanner in a graph $G=(V,E)$ is a subgraph $H=(V,E')$ such that $\delta_H(u,v)\leq t\cdot\delta_G(u,v)$ for all pairs of vertices $u,v\in V$. That is, the stretch t is the maximum distortion between the graph distances δ_G and δ_H. Importantly, when G changes (under edge/vertex insertions or deletions), the underlying metric δ_G changes, as well. The distance $\delta_G(u,v)$ may dramatically decrease upon the insertion of the edge uv. In contrast, our model assumes that the distances in the underlying metric space $\mathcal{M}=(P,\delta)$ remain fixed, but the algorithm can only see the distances between the points that have been presented. For this reason, our results are not directly comparable to models where the underling graph changes dynamically.

For *unweighted* graphs with n vertices, the current best fully dynamic and single-pass streaming algorithms can maintain spanners that achieve almost the same stretch-sparsity trade-off available for the static case: $2k-1$ stretch and $O(n^{1+\frac{1}{k}})$ edges, for $k\geq 1$, which is attained by the greedy algorithm [4], and conjectured to be optimal due to the Erdős girth conjecture [28]. 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. See [7, 9, 11, 16] for some excellent work on dynamic spanners. In the streaming model the input is a sequence (or stream) of edges representing the edge set E of the graph G. A (single-pass) streaming algorithm decides, for each newly arriving edge, whether to include it in the spanner. The graph G is too large to fit in memory, and the objective is to optimize work space and update time [6, 8, 26, 29, 30, 49].

1.2.2 Incremental Algorithms for Geometric Spanners

We briefly review three previously known incremental $(1+\varepsilon)$-spanner algorithms in Euclidean d-space from the perspective of competitive analysis.

Deformable Spanners. Gao et al. [33] designed a dynamic DefSpanner algorithm that maintains a $(1+\varepsilon)$-spanner for a dynamic set S in the Euclidean d-space. For point insertions, it only adds new edges, so it is an online algorithm, as well. It maintains a $(1+\varepsilon)$-spanner with $O_d(\varepsilon^{-d}) \cdot n$ edges and $O_d(\varepsilon^{-(d+1)} \log n)$ lightness. Since the $\|\text{MST}(S)\|$ is a lower bound for the optimal spanner weight, its competitive ratio is also $O_d(\varepsilon^{-(d+1)} \log n)$. The key ingredient of DefSpanner is *hierarchical nets* [42, 46, 55], a form of hierarchical clustering, which can be maintained dynamically. Hierarchical nets naturally generalize to doubling spaces, and so DefSpanner also maintains a $(1+\varepsilon)$-spanner with $\varepsilon^{-O(d)} \cdot n$ edges and $\varepsilon^{-O(d)} \cdot \log n$ lightness for doubling dimension d [35, 55].

Well-Separated Pair Decomposition (WSPD). Well-separated pair decomposition was introduced by Callahan and Kosaraju [21] (see also [37, 41, 51, 58]). For a set S in a metric space, a WSPD is a collection of unordered pairs $W = \{\{A_i, B_i\} : i \in I\}$ such that (1) $A_i, B_i \subset S$ for all $i \in I$; (2) $\min\{\|ab\| : a \in A_i, b \in B_i\} \leq \varrho \cdot \max\{\text{diam}(A_i), \text{diam}(B_i)\}$ for all $i \in I$, where ϱ is the *separation ratio*; (3) for each point pair $\{a, b\} \subset S$ there exists a pair $\{A_i, B_i\}$ such that A_i and B_i each contain one of a and b. Given a WSPD with separation ratio $\varrho > 4$, any graph that contains at least one edge between A_i and B_i, for all $i \in I$, is a spanner with stretch $t = 1 + 8/(\varrho - 4)$. Setting $\varrho \geq 12\varepsilon^{-1}$ for $0 < \varepsilon < 1$, we obtain $t \leq 1 + \varepsilon$.

Hierarchical clustering provides a WSPD [41, Ch. 3]. Perhaps the simplest hierarchical subdivisions in $\mathbb{R}^d$ are quadtrees. Let $\mathcal{T}$ be a quadtree for a finite set $S \subset \mathbb{R}^d$. The root of $\mathcal{T}$ is an axis-aligned cube of side length a_0, which contains S; it is recursively subdivided into 2^d congruent cubes until each leaf cube contains at most one point in S. For all pairs of cubes $\{Q_1, Q_2\}$ at level ℓ of $\mathcal{T}$, create a pair $\{A_i, B_i\}$ with $A_i = Q_1 \cap S$ and $B_i = Q_2 \cap S$ whenever $D_\ell \leq \text{dist}(Q_1, Q_b) < 2D_\ell$ for $D_\ell = \varrho \cdot \text{diam}(Q_1) = 12\varepsilon^{-1} \cdot \sqrt{d} \cdot a_0/2^\ell$; and repeat for all levels $\ell \geq 0$. Properties (1)–(3) of a WSPD are easily verified [41, Ch. 3]. The resulting $(1+\varepsilon)$-spanner has $O_d(\varepsilon^{-d}) \cdot n$ edges [41, 42] and lightness $O_d(\varepsilon^{-(d+1)} \log n)$ [14].

For point insertions in $\mathbb{R}^d$, a dynamic quadtree only adds nodes, which in turn creates new pairs in the WSPD, and new edges in the spanner. This is an online algorithm with the same guarantees as DefSpanner [14, 42] (see also [32] for an efficient implementation).

Ordered Yao-Graphs and Θ-Graphs. Among the first constructions for (offline) sparse $(1+\varepsilon)$-spanners in the Euclidean d-space were Yao- and Θ-graphs [24, 45, 56]. Incremental versions of Yao-graphs and Θ-graphs were introduced by Bose et al. [20]. Let $S = \{s_1, \ldots, s_n\}$ be an ordered set of points in $\mathbb{R}^2$. For each $s_i \in S$, partition the plane into k cones with apex s and aperture $2\pi/k$. The *ordered Yao-graph* $Y_k(S)$ contains an edge between s_i and a closest *previous* point in $\{s_j : j < i\}$ in each cone. The graph $\Theta_k(S)$ is defined similarly, but in each cone the distance to the apex is measured by the orthogonal projection to a ray within the cone. Bose et al. [20] showed that the ordered Yao- and Θ-graphs have spanning ratio at most $1/(1-2\sin(\pi/k))$ for $k > 8$; tighter bounds were later obtained in [19]. In particular, the ordered Yao- and Θ-graphs are $(1+\varepsilon)$-spanners for $k \geq \Omega(\varepsilon^{-1})$.

The construction generalizes to $\mathbb{R}^d$ for all $d \in \mathbb{N}$ [56]. For an angle $\alpha \in (0, \pi)$, let $A \subset \mathbb{S}^{d-1}$ be a maximal set of points in the $(d-1)$-sphere such that $\min_{a,b\in A} \mathrm{dist}(a, b) \leq \alpha$ (in radians). A standard volume argument shows that $|A| \leq O_d(\alpha^{1-d})$. For each $a_i \in A$, create a cone C_i with apex at the origin o, aperture α, and symmetry axis oa_i. Note that $\mathbb{R}^d \subseteq \bigcup_i C_i$. Given a finite set $P \subset \mathbb{R}^d$, we translate each cone C_i to a cone $C_i(p)$ with apex $p \in P$. For every cone $C_i(p)$, the Yao-graph contains an edge between p and a closest point in $P \cap C_i(p)$. For every $\varepsilon > 0$ and $d \in \mathbb{N}$, there exists an angle $\alpha = \alpha(d, \varepsilon) = \Theta_d(\varepsilon)$ for which the Yao-graph is a $(1+\varepsilon)$-spanner for every finite set $P \subset \mathbb{R}^d$.

Ordered Yao- and Θ-graphs give online algorithms for maintaining a $(1+\varepsilon)$-spanner for a sequence of points in $\mathbb{R}^d$. The sparsity of these spanners is bounded by the number of cones per vertex, $O_d(\varepsilon^{1-d})$, which matches the (offline) lower bound of $\Omega_d(\varepsilon^{1-d})$ [47]. However, their weight may be significantly higher than optimal: For n equally spaced points in a unit circle, in any order, Yao- and Θ-graphs yield $(1+\varepsilon)$-spanners of weight $\Omega(\varepsilon^{-1}) \cdot n$, hence lightness $\Omega(\varepsilon^{-1}) \cdot n$, while the optimum weight is $O(\varepsilon^{-2})$ [47].

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 the input sequence of points, but are present in the metric space. An online algorithm is allowed *add* Steiner points, however, the spanner must achieve the given stretch factor only for the input point pairs. It has been observed through a series of work in recent years, that Steiner points allow for substantial improvements over the bounds on the sparsity and lightness of Euclidean spanners in the offline settings . Highly nontrivial insights are required to argue the bounds for Steiner spanners, and often they tend to be even more intricate than their non-Steiner counterpart; see [12, 13, 47, 48]. Bhore and Tóth [14] showed that if an algorithm can use Steiner points, then the competitive ratio for weight improves to $O(\varepsilon^{(1-d)/2} \log n)$ in the Euclidean d-space.

2 Upper Bounds in Euclidean Spaces

We present an online algorithm for a sequence of n points in the Euclidean d-space (Section 2.1). It combines features from several previous approaches, and maintains a $(1+\varepsilon)$-spanner of lightness $O_d(\varepsilon^{-d} \log n)$ and sparsity $O_d(\varepsilon^{1-d} \log \varepsilon^{-1})$ for $d \geq 1$. Lightness is an upper bound for the competitive ratio for weight; the sparsity almost matching the optimal bound $O_d(\varepsilon^{1-d})$ attained by ordered Yao-graphs. In the plane $(d = 2)$, we show that the same algorithm achieves competitive ratio $O(\varepsilon^{-3/2} \log \varepsilon^{-1} \log n)$ using a tighter analysis: A charging scheme that charges the weight of the online spanner to a minimum weight spanner (Section 2.2).

2.1 An Improvement in All Dimensions

We combine features from two incremental algorithms for geometric spanners, and obtain an online $(1+\varepsilon)$-spanner algorithm for a sequence of n points in $\mathbb{R}^d$. We maintain a dynamic quadtree for hierarchical clustering, and use a modified ordered Yao-graph in each level of the hierarchy. In particular, we limit the weight of the edges in the Yao-graph in each level of the hierarchy (thereby avoiding heavy edges). We start with an easy observation.

▶ **Lemma 1.** *Let $G = (S, E)$ be a t-spanner and let $w > 0$. Let $G' = (S, E')$, where $E' = \{e \in E : \|e\| \leq w\}$ is the set of edges of weight at most w. Then for every $a, b \in S$ with $\|ab\| < w/t$, graph G' contains an ab-path of weights at most $t\,\|ab\|$.*

Proof. Since G is a t-spanner, it contains an ab-path P_{ab} of weight at most $t\,\|ab\| \leq w$. By the triangle inequality, every edge in this path has weight at most w, hence present in G'. Consequently G' contains P_{ab}. ◀

The input is a sequence of points $(s_1, s_2, \ldots)$ in $\mathbb{R}^d$, $d \geq 1$. 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 $\mathcal{T}_n$ for S_n. Every node of $\mathcal{T}_n$ corresponds to a cube. The root of $\mathcal{T}_n$, at level 0, corresponds to a cube Q_0 of side length $a_0 = \Theta(\mathrm{diam}(S_n))$. At every level $\ell \geq 0$, there are at most $2^{d\ell}$ interior-disjoint cubes, each of side length $a_\ell = a_0\, 2^{-\ell}$. A cube $Q \in \mathcal{T}_n$ is *nonempty* if $Q \cap S_n \neq \emptyset$. For every nonempty cube Q, we maintain a representative $s(Q) \in Q \cap S_n$, selected at the time when Q becomes nonempty. At each level ℓ, let P_ℓ be the sequence of representatives, in the order in which they are created.

For each level ℓ, we maintain a modified ordered Yao-graph $G_\ell = (P_\ell, E_\ell)$ as follows. When a new point p is inserted into P_ℓ, cover $\mathbb{R}^d$ with $\Theta_d(\varepsilon^{1-d})$ cones of aperture $\alpha(d, \varepsilon)$ as in the construction of Yao-graphs. In each cone C_i, find a point $q_i \in C_i \cap P_\ell$ closest to p; and add pq_i to E_ℓ if $\|pq_i\| < 24 a_\ell \sqrt{d} \cdot \varepsilon^{-1}$. The algorithm maintains the spanner $G = \bigcup_{\ell \geq 0} G_\ell$.

▶ **Theorem 2.** *Let $d \geq 1$ and $\varepsilon \in (0, 1)$. The online algorithm ALG_1 maintains, for a sequence of n points in Euclidean d-space, an $(1 + O(\varepsilon))$-spanner with weight $O_d(\varepsilon^{-d} \log n) \cdot \|MST\|$ and $O_d(\varepsilon^{1-d} \log \varepsilon^{-1}) \cdot n$ edges.*

Note that Theorem 2 implies that the competitive ratio of this algorithm is also $O_d(\varepsilon^{-d} \log n)$.

Proof.

Stretch Analysis. We give a bound on the stretch factor in two steps: First, we define an auxiliary graph $H = (S, E')$ which is a $(1+\varepsilon)$-spanner for S by the analysis of WSPDs. Then we show that G contains an ab-path of weight at most $(1+\varepsilon)\|ab\|$ for each edge of H. Overall, the stretch of G is at most $(1+\varepsilon)^2 = (1 + O(\varepsilon))$ for all $a, b \in S$.

First Layer: WSPD. For each level $\ell \geq 0$, let $H_\ell = (P_\ell, E'_\ell)$ be the graph that contains an edge between two representatives $a, b \in P_\ell$ whenever $\|ab\| \leq 12 a_\ell \sqrt{d} \cdot \varepsilon^{-1}$. Let $H = \bigcup_{\ell \geq 0} H_\ell$. The auxiliary graph H_ℓ contains an edge between the representatives of any such pair of cubes at level ℓ. As noted Section 1.2.2, $H = \bigcup_{\ell \geq 0} H_\ell$ is a $(1+\varepsilon)$-spanner (cf. [41, 42]).

Second Layer: Near-Sighted Yao-graphs. As H is a $(1+\varepsilon)$-spanner, for every $a, b \in S_n$, it contains an ab-path of weight at most $(1+\varepsilon)\|ab\|$. Consider such a path $P_{ab} = (a = p_0, \ldots, p_m = b)$. Each edge $p_{i-1}p_i$ is in H_ℓ for some $\ell \geq 0$. By construction, every edge in H_ℓ has weight at most $12 a_\ell \sqrt{d} \cdot \varepsilon^{-1}$. For every level ℓ, the ordered Yao-graph $Y(P_\ell)$ with angle $\alpha(d, \varepsilon)$ is a $(1+\varepsilon)$-spanner. The graph $G_\ell = (P_\ell, E_\ell)$ constructed by ALG_1 at level ℓ is a subgraph of $Y(P_\ell)$. By Lemma 1, for every $p, q \in P_\ell$ with $\|pq\| \leq 12 a_\ell \sqrt{d} \cdot \varepsilon^{-1}$, graph G_ℓ contains a pq-path of weight at most $(1+\varepsilon)\|pq\|$.

Overall, H contains an ab-path $P_{ab} = (p_0, \ldots, p_m)$ of weight at most $(1+\varepsilon)\|ab\|$. For each edge $p_{i-1}p_i$ of P_{ab}, graph G contains a $p_{i-1}p_i$-path of weight $(1+\varepsilon)\|p_{i-1}p_i\|$. The concatenation of these paths is an ab-path of weight $(1+\varepsilon)^2\|ab\| \leq (1 + O(\varepsilon))\|ab\|$.

Weight Analysis. We may assume w.l.o.g. that the root of the quadtree $\mathcal{T}_n$ is the unit cube $[0,1]^d \subset \mathbb{R}^d$, which has diameter $\sqrt{d}$. This implies $\mathrm{diam}(S_n) \le \sqrt{d} = O_d(1)$. Assume further that $n > 1$, and $\frac{1}{4} \le \mathrm{diam}(S_n) \le \|MST(S_n)\|$.

Every edge in E_ℓ at level ℓ has weight $O_d(\varepsilon^{-1}\,2^{-\ell})$. In particular, every edge at level $\ell \ge 2\log n$ has weight $O_d(\varepsilon^{-1}/n^2)$; and the total weight of these edges is $O_d(\varepsilon^{-1}) \le O_d(\varepsilon^{-1}\|MST(S_n)\|)$.

It remains to bound the weight of the edges on levels $\ell = 1, \ldots, \lfloor 2\log n \rfloor$. At level ℓ of the quadtree $\mathcal{T}_n$, there are at most $2^{d\ell}$ nodes, hence $|P_\ell| \le 2^{d\ell}$. If $|P_\ell| < 3^d$, then G_ℓ has at most $O(3^{2d}) = O_d(1)$ edges, each of weight at most $\mathrm{diam}(P_\ell) \le \mathrm{diam}(S_n) \le \|\mathrm{MST}(S_n)\|$, and so $\|E_\ell\| \le O_d(\|\mathrm{MST}(S_n)\|)$. Assume now that $|G_\ell| \ge 3^d$. By the definition of ordered Yao-graphs, each vertex inserted into P_ℓ adds $\Theta(\varepsilon^{1-d})$ new edges, each of weight $O(\varepsilon^{-1}\,2^{-\ell})$. The total weight of the edges in G_ℓ is at most

$$\|E_\ell\| \le |P_\ell| \cdot \varepsilon^{1-d} \cdot \max_{e \in E_\ell} \|e\| \le O_d(|P_\ell|\,\varepsilon^{-d}\,2^{-\ell}). \tag{1}$$

We next derive a lower bound for $\|\mathrm{MST}(S_n)\|$ in terms of $|P_\ell|$, when $|P_\ell| > 1$ and $\ell > 2$, using a standard volume argument. Define a graph on the vertex set P_ℓ such that two nodes $p, q \in P_\ell$ are adjacent iff p and q lie in neighboring quadtree cells of level ℓ. Since every quadtree cell has $3^d - 1$ neighbors, this graph is $(3^d - 1)$-degenerate, and contains an independent set I_ℓ of size at least $(3^d - 1)^{-1}|P_\ell| = \Omega_d(|P_\ell|)$. The distance between any two disjoint quadtreee cells at level ℓ is at least $2^{-\ell}$. Consequently, the open balls of radius $2^{-(\ell+1)}$ centered at the points in I_ℓ are pairwise disjoint. None of the balls contains S_n for $\ell > 2$, as the diameter of each of ball is $2^{-\ell}$ while $\mathrm{diam}(S_n) \ge \frac{1}{4}$. For all $\ell > 2$, $\mathrm{MST}(S_n)$ contains the center of each ball and a point in its exterior; hence the intersection of $\mathrm{MST}(S_n)$ and each ball contains a path from the center to a boundary point, which has weight at least $2^{-(\ell+1)}$. Summation over $|I_\ell|$ disjoint balls yields

$$\|MST(S_n)\| \ge |I_\ell| \cdot 2^{-(\ell+1)} \ge \Omega_d(|P_\ell|\,2^{-\ell}). \tag{2}$$

Comparing inequalities (1) and (2), we obtain $\|E_\ell\| \le O_d(\varepsilon^{-d}) \cdot \|MST(S_n)\|$. Summation over all levels $\ell \in \mathbb{N}$ yields $\|E\| \le O_d(\varepsilon^{-d}\log n) \cdot \|\mathrm{MST}(S_n)\|$, as claimed.

Sparsity Analysis. In the full paper, we show that G has $O(\varepsilon^{1-d}\log\varepsilon^{-1}) \cdot n$ edges. ◀

2.2 Further Improvements in the Plane

We presents a tighter analysis of algorithm ALG_1 for $d = 2$ that compares the spanner weight to the offline optimum weight, and bypasses the comparison with the MST (i.e., lightness).

Minimum-Weight Euclidean $(1+\varepsilon)$-Spanner. For any $a, b \in \mathbb{R}^d$, an ab-path P_{ab} of Euclidean weight at most $(1+\varepsilon)\|ab\|$ lies in the ellipsoid $\mathcal{E}_{ab}$ with foci a and b and great axes of weight $(1+\varepsilon)\|ab\|$; see Figure 1. A key observation is that the minor axis of $\mathcal{E}_{ab}$ is $((1+\varepsilon)^2 - 1^2)^{1/2}\,\|ab\| \approx \sqrt{2\varepsilon}\,\|ab\|$. Furthermore, Bhore and Tóth [13] recently observed that the directions of "most" edges of the path P_{ab} are "close" to the direction of ab. Specifically, if we denote by $E(\alpha)$ the set of edges e in P_{ab} with $\angle(ab, e) \le \alpha$, then the following holds.

▶ **Lemma 3** (Bhore and Tóth [13]). *Let $a, b \in \mathbb{R}^d$ and let P_{ab} be an ab-path of weight $\|P_{ab}\| \le (1+\varepsilon)\|ab\|$. Then for every $i \in \{1, \ldots, \lfloor 1/\sqrt{\varepsilon} \rfloor\}$, we have $\|E(i \cdot \sqrt{\varepsilon})\| \ge (1 - 2/i^2)\,\|ab\|$.*

Let $R(a,b) = \mathcal{E}_{ab} \cap \mathcal{N}(a,b)$, where $\mathcal{N}(a,b)$ is the annulus bounded by two concentric spheres centered at a, of radii $\frac{1+\varepsilon}{2}\,\|ab\|$ and $\|ab\|$; see Figure 1 for an example.

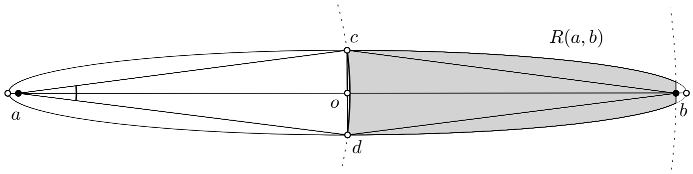

Figure 1 Any ab-path of weight at most $(1+\varepsilon)\|ab\|$ lies in the ellipse $\mathcal{E}_{ab}$ with foci a and b. The shaded region $R(a,b)$ is the part of the ellipse $\mathcal{E}_{ab}$ between two concentric circles centered at a.

▶ **Lemma 4.** *If $0 < \varepsilon < \frac{1}{9}$, then every ab-path P_{ab} of weight at most $\|P_{ab}\| \leq (1+\varepsilon)\|ab\|$ contains interior-disjoint line segments $s \subset R(a,b)$ of total weight at least $\frac{1}{9}\|ab\|$ such that $\angle(\overrightarrow{ab}, s) \leq 3 \cdot \sqrt{\varepsilon}$.*

Proof. Since the distance between the two concentric circles is $\frac{1-\varepsilon}{2}\|ab\|$, every ab-path contains a subpath of weight at least $\frac{1-\varepsilon}{2}\|ab\|$ in the annulus $\mathcal{N}(a,b)$.

Let P_{ab} be an ab-path of weight at most $(1+\varepsilon)\|ab\|$. As noted above $P_{ab} \subset \mathcal{E}_{ab}$. Hence, $\|P_{ab} \cap \mathcal{N}(a,b)\| = \|P_{ab} \cap R(a,b)\| \geq \frac{1-\varepsilon}{2}\|ab\|$ in $R(ab)$; and so $\|P_{ab} \setminus R(a,b)\| = \|P_{ab}\| - \|P_{ab} \cap R(a,b)\| \leq \frac{1+3\varepsilon}{2}\|ab\|$.

Applying Lemma 3 with $i = 3$, the total weight of the edges e of P_{ab} with $\mathrm{dir}(ab, e) \leq 3 \cdot \sqrt{\varepsilon}$ is at least $\frac{7}{9}\|ab\|$. The parts of these edges lying outside of $R(a,b)$ have weight at most $\|P_{ab} \setminus R(a,b)\| \leq \frac{1+3\varepsilon}{2}\|ab\|$. Consequently, the remaining part of these edges are in $R(a,b)$, and their weight is at least $\left(\frac{7}{9} - \frac{1+3\varepsilon}{2}\right)\|ab\| = \frac{5-27\varepsilon}{18}\|ab\| > \frac{1}{9}\|ab\|$ if $\varepsilon < \frac{1}{9}$, as claimed ◀

We also need an observation from elementary geometry; see Figure 1.

▶ **Lemma 5.** *For $a, b \in \mathbb{R}^d$, let cd be the minor axis of the ellipsoid $\mathcal{E}_{ab}$. Then $\angle cad \leq \sqrt{8\varepsilon}$.*

Proof. We may assume w.l.o.g. that $\|ab\| = 1$. Let o be the center of the ellipsoid $\mathcal{E}_{ab}$. Then $\sec \angle cao = (\cos \angle cao)^{-1} = \frac{\|ac\|}{\|ao\|} = 1+\varepsilon$. The Taylor estimate $\sec(x) = 1 + \frac{1}{2}x^2 + \frac{5}{24}x^4 + \ldots \geq 1 + \frac{1}{2}x^2$ for $0 < x < 1$ yields $\angle cao \leq \sqrt{2\varepsilon}$. Consequently, $\angle cad = 2\angle cao \geq \sqrt{8\,\varepsilon}$. ◀

▶ **Theorem 6.** *Let $d = 2$ and $\varepsilon \in (0,1)$. The online algorithm* ALG_1 *maintains, for a sequence of n points in Euclidean plane, an $(1+\varepsilon)$-spanner of weight $O(\varepsilon^{-3/2} \log \varepsilon^{-1} \log n) \cdot \mathrm{OPT}$, where* OPT *denotes the minimum weight of an $(1+\varepsilon)$-spanner for the same point set.*

Proof. Theorem 2 has established that algorithm ALG_1 maintains a $(1+\varepsilon)$-spanner. The tighter competitive analysis uses Lemmas 4 and 5.

Competitive Analysis. Assume w.l.o.g. that $\mathrm{diam}(S_n) = \Theta(1)$, hence the side length of every quadtree square at level ℓ is $\Theta(2^{-\ell})$. For a set $S_n = \{s_1, \ldots, s_n\} \subset \mathbb{R}^2$, let $G^* = (S_n, E^*)$ be a $(1+\varepsilon)$-spanner of minimum weight, and let $\mathrm{OPT} = \|G^*\|$. Let $G = (S_n, E)$ be the spanner returned by the online algorithm ALG_1. Recall that $G = \bigcup_{\ell \geq 0} G_\ell$, where the total weight of all edges at levels $\ell > 2\log n$ is less than $\mathrm{diam}(S_n)$, so it is enough to consider $\ell = 0, \ldots, \lceil 2\log n \rceil$.

▷ Claim 7. $\|G_\ell\| \leq O(\varepsilon^{-3/2} \log \varepsilon^{-1}) \cdot \mathrm{OPT}$ for all $\ell \geq 0$.

Claim 7 immediately implies $\|G\| \leq O(\varepsilon^{-3/2} \log \varepsilon^{-1} \log n) \cdot \mathrm{OPT}$. For every level $\ell \geq 0$, $G_\ell = (P_\ell, E_\ell)$ is a graph on the representatives P_ℓ. Note that G^* is a Steiner spanner with respect to the point set P_ℓ, as G^* is a spanner on all n points of the input.

We prove Claim 7 using a charging scheme: We charge the weight of every edge in G_ℓ to G^* (more precisely, to line segments along the edges of G^*), and then show that each line segment of weight w in G^* receives $O(\varepsilon^{-3/2} \log \varepsilon^{-1}) \cdot w$ charge. For every point $p \in P_\ell$, algorithm ALG_1 greedily covers $\mathbb{R}^2$ by $\Theta(\varepsilon^{-1})$ cones of aperture $\pi/k = \Theta(\varepsilon^{-1})$ and apex p, and adds an edge pq_i in each nonempty cone C_i. For the competitive analysis, we greedily cover $\mathbb{R}^2$ by $\Theta(\varepsilon^{-1/2})$ cones of aperture $\sqrt{\varepsilon}$ and apex p. We use translates of the same cone cover for all $p \in P_\ell$. Standard volume argument implies that a cone of aperture $\sqrt{\varepsilon}$ intersects $O(\varepsilon^{-1/2})$ cones of aperture $\Theta(\varepsilon^{-1})$. We describe the charging scheme for each such cone $\widehat{C}$.

Charging Scheme. Consider a cone $\widehat{C}$ with apex p and aperture $\sqrt{\varepsilon}$. Let $E(\widehat{C})$ be the set of edges pq, $q \in \widehat{C}$ that algorithm ALG_1 adds to G_ℓ when p is inserted into P_ℓ. Since $\widehat{C}$ intersects $O(\varepsilon^{-1/2})$ cones of the ordered Yao-graph, then $|E(\widehat{C})| \leq O(\varepsilon^{-1/2})$. By construction, every edge in G_ℓ has weight at most $O(\varepsilon^{-1}2^{-\ell})$. Hence

$$\|E(\widehat{C})\| = \sum_{pq \in E(\widehat{C})} \|pq\| \leq |E(\widehat{C})| \cdot O(\varepsilon^{-1}2^{-\ell}) \leq O(\varepsilon^{-3/2}2^{-\ell}). \quad (3)$$

Let $q_0 = q_0(\widehat{C})$ be a closest point in $P_\ell \cap \widehat{C}$ to p. (Possibly, q_0 arrived after p.) We distinguish between two cases:

Case 1: $\|pq_0\| < 2 \cdot 2^{-\ell}$. Since $q_0 \in P_\ell$, and P_ℓ contains at most one point in each quadtree cell of side length $\Theta(2^{-\ell})$, this case occurs at most $O(1)$ times per apex p. On the one hand, the summation of (3) over all $p \in P_\ell$ and all cones $\widehat{C}$ with $\|pq_0\| < 2 \cdot 2^{-\ell}$ is bounded by $O(|P_\ell| \cdot \varepsilon^{-3/2}2^{-\ell})$. On the other hand, $\mathrm{OPT} \geq \Omega(\|\mathrm{MST}(P_\ell)\|) \geq \Omega(|P_\ell| \cdot 2^{-\ell})$. Consequently, the total weight of all edges handled in Case 1 is $O(\varepsilon^{-3/2})\,\mathrm{OPT}$.

Case 2: $\|pq_0\| \geq 2 \cdot 2^{-\ell}$. The optimal spanner G^* contains a pq_0-path P_0 of weight at most $(1+\varepsilon)\|pq_0\|$. Recall P_0 lies in the ellipse $\mathcal{E}_0$ with foci p and q_0, and $R(p, q_0)$ is the half of $\mathcal{E}_0$ that contains q_0 (cf. Figure 1). Let $E^*(\widehat{C})$ be the set of maximal line segments e along edges in E^* such that $e \subset P_0 \cap R(p, q_0)$ and $\angle(e, pq_0) \leq 3 \cdot \sqrt{\varepsilon}$. By Lemma 4, we have $\|E^*(\widehat{C})\| \geq \frac{1}{9}\|pq_0\|$. We distribute the weight of all edges in $E(\widehat{C})$ uniformly among the line segments in $E^*(\widehat{C})$. That is, each segment of weight w in $E^*(\widehat{C})$ receives a charge of

$$\frac{\|E(\widehat{C})\|}{\|E^*(\widehat{C})\|} \cdot w \leq \frac{O(\varepsilon^{-3/2}2^{-\ell})}{\Omega(2^{-\ell})} \cdot w \leq O(\varepsilon^{-3/2}) \cdot w. \quad (4)$$

This completes the description of the charging scheme in Case 2.

Charges Received. A point along an edge of the optimal spanner G^* may receive charges from several cones $\widehat{C}$, possibly with different apices $p \in P_\ell$. Let L be a maximal line segment along an edge of G^* such that every point in L receives the same charges.

For a cone $\widehat{C}$ of aperture $\sqrt{\varepsilon}$, let $\widehat{K}$ denote a cone with the same apex and axis as $\widehat{C}$, but aperture $3\sqrt{\varepsilon}$; refer to Figure 2.

▷ Claim 8. If L receives charges from $\widehat{C}$, then $L \subset \widehat{K}$.

Indeed, if L receive charges from $\widehat{C}$, then $L \subset R(p, q_0) \subset E_0$, where $\mathcal{E}_0$ is the ellipse with foci p and the closest point $q_0 \in \widehat{C} \cap P_\ell$. By Lemma 5, $R(p, q_0)$ lies in a cone with apex p, aperture $2\sqrt{\varepsilon}$, and axis pq_0. Consequently $L \subset R(p, q_0) \subset \widehat{K}$, which proves Claim 8.

Note that if L receives positive charge from a cone $\widehat{C}$ with apex p and closest point q_0, then $\angle(L, pq_0) \leq 3 \cdot \sqrt{\varepsilon}$. Since the aperture of the cones $\widehat{C}$ is $\sqrt{\varepsilon}$, then L receives charges from cones $\widehat{C}$ with at most $O(1)$ different orientations. We may restrict ourselves to cones $\widehat{C}$ that are translates of each other (but have different apices in P_ℓ).

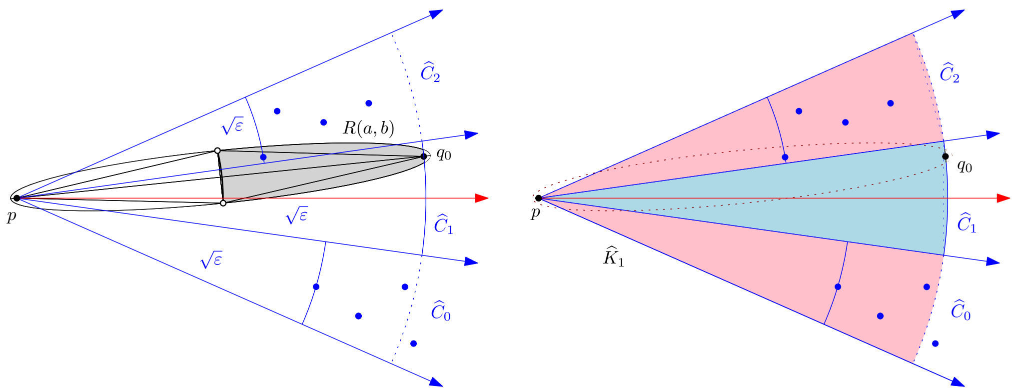

Figure 2 Left: There consecutive cones, $\widehat{C}_0$, $\widehat{C}_1$, and $\widehat{C}_1$, with apex p and aperture $\sqrt{\varepsilon}$. Point q_0 is the closest to p in $P_\ell \cap \widehat{C}_1$; and $R(p, q_0) \subset \widehat{K}_1 = \widehat{C}_0 \cup \widehat{C}_1 \cup \widehat{C}_2$. Right: No point in P_ℓ is in the blue sector $\widehat{K}$, but there may be points in the pink sectors.

Let $\mathcal{A}$ be the set of all translates of a cone $\widehat{C}$ with aperture $\sqrt{\varepsilon}$ and apices in P_ℓ, and L receives positive charge from $\widehat{C}$. We partition $\mathcal{A}$ into $O(\log \varepsilon^{-1})$ classes as follows. For $j = 1, \ldots, \lceil \log(2\varepsilon^{-1}) \rceil$, let $\mathcal{A}_j$ be the set of cones $\widehat{C} \in \mathcal{A}$ such that $2^{j-\ell} \leq \|pq_0\| < 2^{j+1-\ell}$, where $p \in P_\ell$ is the apex of $\widehat{C}$ and q_0 is the closest point in $P_\ell \cap \widehat{C}$ to p.

▷ Claim 9. For each j, segment L receives $O(\varepsilon^{-3/2})\,\|L\|$ total charges from all cones in $\mathcal{A}_j$.

For a cone $\widehat{C} \in \mathcal{A}_j$, the bound (3) is replaced by

$$\|E(\widehat{C})\| = \sum_{pq \in E(\widehat{C})} \|pq\| \leq |E(\widehat{C})| \cdot O(2^{j-\ell}) \leq O(\varepsilon^{-1/2} 2^{j-\ell}), \tag{5}$$

while $\|E^*(\widehat{C})\| \geq \frac{1}{9}\|pq_0\| \geq \Omega(2^{j-\ell})$ by Lemma 4. Refining (4), L receives a charge

$$\frac{\|E(\widehat{C})\|}{\|E^*(\widehat{C})\|} \cdot \|L\| \leq \frac{O(\varepsilon^{-1/2} 2^{j-\ell})}{\Omega(2^{j-\ell})} \cdot \|L\| \leq O(\varepsilon^{-1/2}) \cdot \|L\| \tag{6}$$

from each cone in $\mathcal{A}_j$. To prove Claim 9, it is enough to show that $|\mathcal{A}_j| \leq O(2^j) \leq O(\varepsilon^{-1})$.

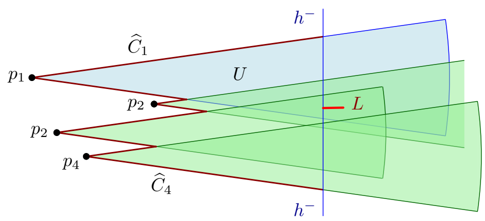

Figure 3 The union U of triangles $\widehat{C} \cap h^-$, where L receives charges from the cones $\widehat{C}$.

By Claim 8, L received charges from cones of $O(1)$ different orientations. We consider each orientation separately. We may assume w.l.o.g. that the symmetry axis of every cone in $\mathcal{A}_j$ is parallel to the x-axis, and their apex is their leftmost point. Let h be a vertical line that contains the left endpoint of L, and let h^- be the left halfplane bounded by h; see Figure 3. The intersections $\widehat{C} \cap h$ and $\widehat{K} \cap h$ are vertical line segments of length $O(2^{j-\ell} \tan \sqrt{\varepsilon})$. We have $L \cap h \subset \widehat{K} \cap h$ by Claim 8; and obviously $\widehat{C} \cap h \subset \widehat{K} \cap h$. Consequently, a vertical line segment of length $O(2^{j-\ell} \tan \sqrt{\varepsilon})$ contains $h \cap \widehat{C}$ for all $\widehat{C} \in \mathcal{A}_j$.

Let U be the union of the triangles $\widehat{C} \cap h^-$ for all $\widehat{C} \in \mathcal{A}_j$. The interior of $\widehat{C} \cap h^-$ does not contain any point in P_ℓ. Consequently, the apices of all cones lie on the boundary ∂U of U. The part of ∂U in h^- is a y-monotone curve with slopes $\pm\sqrt{\varepsilon}$. It follows that the length of ∂U is $O(2^{j-\ell} \tan\sqrt{\varepsilon}/\sin\sqrt{\varepsilon}) = O(2^{j-\ell} \csc\sqrt{\varepsilon}) = O(2^{j-\ell})$. This, in turn, implies that ∂U intersects $O(2^j)$ cubes of side length $a_0 2^{-\ell}$ at level ℓ of the quadtree, and so $|\mathcal{A}_j| \le O(2^j) \le O(\varepsilon^{-1})$, as required. This completes the proof Claim 9, and hence the proof of Theorem 6. ◀

3 Lower Bounds in $\mathbb{R}^d$ Under the L_1 Norm

In this section we introduce a strategy based on the points on the integer lattice $\mathbb{Z}^d$, that achieves a new lower bound for the competitive ratio of an online $(1+\varepsilon)$-spanner algorithm in $\mathbb{R}^d$ under the L_1 norm.

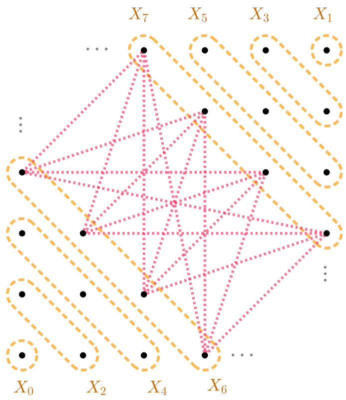

Figure 4 A sketch of the construction for the lower bound in two dimensions. Any online algorithm is required to add the red pairs.

Construction. We describe an adversary strategy with $\Omega_d(\varepsilon^{-d})$ points and show that any online algorithm returns a $(1+\varepsilon)$-spanner whose weight is $\Omega_d(\varepsilon^{-d})$ times the optimum weight. One can extend this result for arbitrary number of points, but that does not necessarily improve the lower bound. The final point set X consists of the points of the integer lattice $\mathbb{Z}^d$ in the hypercube $[0, \frac{1}{\varepsilon d})^d$, where $\varepsilon < \frac{1}{d}$. The points are presented in stages in order to deceive the online algorithm to add more edges than needed. In step $2i$, where $0 \le i < \frac{1}{2\varepsilon}$, points $x \in X$ such that $\|x\|_1 = i$ will be given to the algorithm. In step $2i+1$, where $0 \le i < \frac{1}{2\varepsilon}$, the adversary presents points $x \in X$ such that $\|x\|_1 = \lceil 1/\varepsilon \rceil - i$ (Figure 4). In other words, points are presented in batches according to their L_1 norms.

Competitive Ratio. Denote by X_i the set of points presented in step i. The idea is to show that there has to exist many edges between X_i and X_{i+1} in order to guarantee the $1+\varepsilon$ stretch-factor. Specifically, we define an *ordered-pair* as follows.

▶ **Definition 10** (ordered-pair). *A pair of points (x, y) in $\mathbb{R}^d$ is an* ordered-pair *if $x \in X_{2i}$ and $y \in X_{2i+1}$ for some i, and $x_k \le y_k$ for all k, where x_k and y_k are the k-th coordinates of x and y respectively.*

Now we show that any ordered-pair $(x, y) \in X_{2i} \times X_{2i+1}$ requires an edge in the spanner immediately after x and y are presented. To prove this, we show (in the full paper) that previously presented points cannot serve as via points in a $(1+\varepsilon)$-path between x and y.

▶ **Lemma 11.** *Let (x, y) be an ordered-pair. Then there is no $(1+\varepsilon)$-path between x and y that goes through any other point $z \in X_j$ with $j \leq i+1$.*

We next show that the total weight of the edges between ordered pairs is $\Omega_d(\varepsilon^{-2d})$.

▶ **Lemma 12.** *The total weight of the edges between the ordered-pairs is $\Omega_d(\varepsilon^{-2d})$.*

Proof. Let $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ be two points in X. We show that if $x_k \in [\frac{1}{4\varepsilon(d+0.25)}, \frac{1}{4\varepsilon d}]$ for all $1 \leq k \leq d$, and $y_k \in [\frac{3}{4\varepsilon(d+0.25)}, \frac{3}{4\varepsilon d}]$ for all $1 \leq k \leq d-1$, then there is a choice of y_d that makes (x, y) an ordered-pair. This would imply that there are $\Omega_d(\varepsilon^{-2d+1})$ ordered-pairs and by Lemma 11, each pair requires an edge of weight $\Omega_d(\varepsilon^{-1})$, thus the total weight of required edges would be $\Omega_d(\varepsilon^{-2d})$.

In order to find such a y_d, recall that $\|x\|_1 + \|y\|_1 = \lceil \varepsilon^{-1} \rceil$ holds because (x, y) is an ordered-pair. This equality uniquely determines the value of y_d,

$$y_d = \lceil \varepsilon^{-1} \rceil - \sum_{k=1}^{d} x_k - \sum_{k=1}^{d-1} y_k.$$

We just need to prove the inequalities $y_k \geq x_k$ and $y_k \leq 1/(\varepsilon d)$ for this unique y_k. This can simply be done by plugging the maximum (and minimum) values of x_ks and other y_ks and calculating the result,

$$y_d \geq \frac{1}{\varepsilon} - \frac{d}{4\varepsilon d} - \frac{3(d-1)}{4\varepsilon d} = \frac{3}{4\varepsilon d} > x_d.$$

Also,

$$y_d \leq \frac{1}{\varepsilon} + 1 - \frac{d}{4\varepsilon(d+0.25)} - \frac{3(d-1)}{4\varepsilon(d+0.25)} = 1 + \frac{1}{\varepsilon(d+0.25)} < \frac{1}{\varepsilon d}.$$ ◀

Now we can prove the main theorem of this section.

▶ **Theorem 13.** *The competitive ratio of any online $(1+\varepsilon)$-spanner algorithm in $\mathbb{R}^d$ under the L_1-norm is $\Omega_d(\varepsilon^{-d})$.*

Proof. For the point set $X \subset \mathbb{R}^d$, the unit-distance graph is a Manhattan network: It contains a path of weight $\|xy\|_1$ for all $x, y \in X$. Its weight is $\Theta_d(\varepsilon^{-d})$ which is an upper bound for the weight of a $(1+\varepsilon)$-spanner for any $\varepsilon \geq 1$. By Lemma 12, any online algorithm returns a spanner of weight $\Omega_d(\varepsilon^{-2d})$. Thus its competitive ratio is $\Omega_d(\varepsilon^{-d})$. ◀

4 General Metrics: The Ordered Greedy Spanner

In this section we study the online spanners problem on general metric spaces. The points arrive one by one, where for each new point we also receive its distances to all previously introduced points.

In the offline setting, the celebrated greedy spanner algorithm [4] sorts the edges by increasing weight, and then processes them one by one, adding each edge if by the time of examination, the distance between its endpoints is too large. This algorithm achieves the existentially optimal[2] sparsity and lightness as a function of the stretch factor [31]. However, in the online model, we do not receive the edges in a sorted order, and therefore cannot execute the greedy algorithm. As an alternative, we propose here the *ordered greedy* algorithm. This is a deterministic algorithm working against an adaptive adversary. The algorithm receives a stretch factor t, and works naturally as follows: We maintain a spanner H. When a point v_i arrives, we order its edges[3] in the original metric by weight. Each edge $\{v_{i'}, v_i\}$ is added to the spanner H if currently $\delta_H(v_{i'}, v_i) > t \cdot \delta_X(v_{i'}, v_i)$. Note that this algorithm can be easily executed in an online fashion.

▶ **Theorem 14.** *Given an n-point metric space (X, δ_X) in an (adaptive) adversarial order, with stretch factor $t = (2k-1)(1+\varepsilon)$ for $k \geq 2$ and $\varepsilon \in (0,1)$, the ordered greedy algorithm returns a spanner with $O(\varepsilon^{-1}\log\varepsilon^{-1}) \cdot n^{1+\frac{1}{k}}$ edges and weight $O(\varepsilon^{-1} n^{\frac{1}{k}} \log^2 n) \cdot w(\mathrm{MST})$.*

Proof. The bounded stretch of our spanner is straightforward by construction, as every pair was examined at some point, and taken care of. Next we analyze the lightness.

In the online spanning tree problem, points of a finite metric space arrive one-by-one, and we need to connect each new point to a previous point to maintain a spanning tree. The ordered greedy algorithm connects each vertex v_i, to the closest vertex in $\{v_1, \dots, v_{i-1}\}$. As was shown by Imase and Waxman [44], the tree created by the ordered greedy algorithm has lightness $O(\log n)$, which is the best possible [44]. Denote the online spanning tree by T_G. Note that the ordered greedy spanner H will contain T_G, as a shortest edge between a new vertex to a previously introduced vertex is always added to the spanner H. The following clustering lemma is frequently used for spanner constructions (see e.g. [3, 22, 27]). We provide a proof for the sake of completeness.

▷ Claim 15. For every $i \in \mathbb{N}$, the point set X can be partitioned into clusters $\mathcal{C}_i$ of diameter at most $D_i = \varepsilon \cdot (1+\varepsilon)^i$ w.r.t. the metric δ_{T_G} such that $|\mathcal{C}_i| = O(\frac{w(T_G)}{\varepsilon \cdot (1+\varepsilon)^i})$.

Proof. Let N_i be a maximal set of vertices such that for every $x, y \in N_i$, $\delta_{T_G}(x,y) > \frac{1}{2} \cdot D_i$. For every vertex $x \in N_i$ let $C_x = \{z : x = \mathrm{argmin}_{y \in N_i} \delta_X(z,y)\}$ be the Voronoi cell of x. Clearly, $\mathrm{diam}(C_x) \leq D_i$ for all x. Further, consider a continuous version of T_G (where each edge is an interval). Then as the graph T_G is connected, each cluster C_x contains at least $\frac{1}{4}D_i$ length of edges (as the balls $\{B_{T_G}(x, \frac{1}{4}D_i)\}_{x \in N_i}$ are pairwise disjoint). It follows that

$$|\mathcal{C}_i| = |N_i| \leq \frac{w(T_G)}{\frac{1}{4}D_i} = O\left(\frac{w(T_G)}{\varepsilon \cdot (1+\varepsilon)^i}\right). \qquad ◁$$

For every i, consider the *scale* $E_i = \{e = \{u,v\} \in H : (1+\varepsilon)^{i-1} \leq \delta_X(u,v) < (1+\varepsilon)^i\}$. We are now ready to bound the lightness and the sparsity of the ordered greedy spanner. This is accomplished in the next two claims, with proofs in the full paper.

▷ Claim 16. The weight of the ordered greedy spanner is $O(n^{\frac{1}{k}} \cdot \varepsilon^{-2} \log^2 n) \cdot w(\mathrm{MST})$.

▷ Claim 17. The ordered greedy spanner has $O(\varepsilon^{-1} \log \frac{1}{\varepsilon}) \cdot n^{1+\frac{1}{k}}$ edges.

This completes the proof of Theorem 14. ◀

---

[2] Specifically, if a t-spanner construction achieves an upper bound $m(n,t)$ and $l(n,t)$, resp., on the size and lightness of an n-vertex graph then this bound also holds for the greedy t-spanner [31].

[3] By edges we mean point pairs in the metric space, we will often use notation from graph theory.

5 Lower Bound for General Metrics

In this section we prove an $\Omega(\frac{1}{k}\cdot n^{\frac{1}{k}})$ lower bound on the competitive ratio of an online $(2k-1)$-spanner of n-vertex graphs. Our lower bound holds in both cases where the quality is measured by number of edges or the weight. It follows that our upper bound in Theorem 14 cannot be substantially improved, even if we consider competitive ratio instead of lightness/sparsity.

Recall that the Erdős Girth Conjecture [28] states that for every $n,k\geq 1$, there exists an n-vertex graph with $\Omega(n^{1+\frac{1}{k}})$ edges and girth $2k+2$. The proof of the following lemma is based on a counting argument form the recent lower bound proof for (static) vertex fault tolerant emulators by Bodwin, Dinitz, and Nazari [15].

▶ **Lemma 18.** *Assuming the Erdős girth conjecture, for every $n,k\geq 1$, there exists an n-point metric space (X,δ_X) with diameter $2k-1$, such that every $(2k-1)$-spanner has $\Omega(\frac{1}{k}\cdot n^{1+\frac{1}{k}})$ edges and weight $\Omega(n^{1+\frac{1}{k}})$.*

Proof. Let $G=(V,E_G)$ be the graph fulfilling the Erdős girth conjecture. That is, G is an unweighted n-vertex graph with girth $2k+2$ and $|E_G|=\Omega(n^{1+\frac{1}{k}})$ edges. Set a metric δ_X over V as follows,[4]

$$\forall u,v\in V\quad \delta_X(u,v)=\min\left\{\delta_G(u,v),2k-1\right\}.$$

Suppose that $H=(V,E_H)$ is a $(2k-1)$-spanner for (V,δ_X) with weight function w_H, where the weight of an edge $e'\in\{u,v\}\in E_H$ is $w_H(e')=\delta_X(u,v)$. Let $E'=E_H\setminus E_G$ be the edges of H which are not in G. We say that an edge $e'\in E'$ *covers* an edge $e\in E_G$, if there is a shortest path in G between the endpoints of e' going through e of weight at most k. Note that as e' has weight at most k, there is a unique shortest path in G between its endpoints. In particular, each edge $e\in E'$ can cover at most k edges in E_G.

Consider an edge $e=\{v_0,v_s\}\in E_G\setminus E_H$. We argue that some edge $e'\in E'$ must cover e. Suppose for contradiction otherwise, and let $P=(v_0,v_1,\dots,v_s)$ be the shortest path in H between the endpoints v_0,v_s of e. Suppose first that P contains an edge v_i,v_{i+1} of weight at least $w_H(\{v_i,v_{i+1}\})\geq k+1$. In particular, $\delta_G(\{v_i,v_{i+1}\})\geq k+1$. Then by the triangle inequality, $\delta_G(v_0,v_i)+\delta_G(v_{i+1},v_s)\geq\delta_G(v_i,v_{i+1})-\delta_G(v_0,v_s)\geq k$. It follows that P has weight at least $2k+1$, a contradiction to the fact that H is a $2k-1$ spanner. We conclude that for every $i\in\{0,\dots,s-1\}$, $\delta_X(v_i,v_{i+1})=\delta_G(v_i,v_{i+1})\leq k$. In particular, in G there is a unique path $P_i=(u^i_0,\dots,u^i_{s_i})$ between v_i to v_{i+1} of weight $\delta_G(v_i,v_{i+1})\leq k$. As no edge covers e, e does not belong to any of these paths. The concatenation of this paths $P_0\circ P_1\circ\cdots\circ P_{s-1}$ is a path in G of at most $2k-1$ edges between the endpoints of e. It follows that G contains a $2k$-cycle, a contradiction.

For conclusion, as every edge in $E_G\setminus E_H$ is covered, and every edge in $E'=E_H\setminus E_G$ can cover at most k edges, it follows that $|E_H\setminus E_G|\geq\frac{1}{k}\cdot|E_G\setminus E_H|$. In particular,

$$|E_H|=|E_H\cap E_G|+|E_H\setminus E_G|\geq|E_H\cap E_G|+\frac{1}{k}\cdot|E_G\setminus E_H|\geq\frac{1}{k}\cdot|E_G|\,.$$

To bound the weight, for each edge $e'=\{s,t\}\in E'$, let $A_{e'}$ be the set of edges in E_G covered by e'. Note that $w_H(e')=\delta_G(s,t)=|A_{e'}|$. As all the edges in $E_G\setminus E_H$ are covered, we conclude

[4] Note that $\forall x,y,z\in V$, $\delta_X(x,z)=\min\left\{\delta_G(x,z),2k-1\right\}\leq\min\left\{\delta_G(x,y)+\delta_G(y,z),2k-1\right\}\leq\min\left\{\delta_G(x,y),2k-1\right\}+\min\left\{\delta_G(y,z),2k-1\right\}=\delta_X(x,y)+\delta_X(y,z)$. Thus δ_X is a metric space.

$$\begin{aligned}w_H(E_H) &= w_H(E_H \cap E_G) + w_H(E_H \setminus E_G) \\ &= |E_H \cap E_G| + \sum_{e' \in E'} |A_{e'}| \\ &\geq |E_H \cap E_G| + |E_G \setminus E_H| = |E_G| = \Omega(n^{1+\frac{1}{k}}).\end{aligned}$$ ◀

▶ **Theorem 19.** *Assuming Erdős girth conjecture, the competitive ratio of any online $(2k-1)$-spanner algorithm for n-point metrics is $\Omega(\frac{1}{k} \cdot n^{\frac{1}{k}})$, for both weight and number of edges. In more details, there is an n-point metric space (X, δ_X) with a $(2k-1)$-spanner $H_{\mathrm{OPT}} = (X, E_{\mathrm{OPT}})$, and order over X for which every $(2k-1)$-spanner produced by an online algorithm will have $\Omega(\frac{1}{k} \cdot n^{\frac{1}{k}}) \cdot |E_{\mathrm{OPT}}|$ edges, and $\Omega(\frac{1}{k} \cdot n^{\frac{1}{k}}) \cdot w(H_{\mathrm{OPT}})$ weight.*

Proof. Consider the metric space (X, δ_X) from Lemma 18 with parameters $n-1$ and k. Let X' be the metric space X with an additional point r at distance $\frac{2k-1}{2}$ from all the points in X. Note that no pairwise distance is changed due to the introduction of r. The adversary provides the online algorithm the points in X first (in some arbitrary order), and the point r last. After the algorithm received all the points in X', it has a $2k-1$-spanner H_{n-1}. According to Lemma 18, H_{n-1} has $\Omega(\frac{1}{k} \cdot (n-1)^{1+\frac{1}{k}}) = \Omega(\frac{1}{k} \cdot n^{1+\frac{1}{k}})$ edges, and $\Omega(n^{1+\frac{1}{k}})$ weight.

Next the algorithm introduces r. Consider the spanner $S = (X', E_S)$ consisting of $n-1$ edges with r as a center. Note that the maximum distance in S is $2k-1$, and hence S is a $2k-1$ spanner as required. Note that S contains $n-1$ edges of weight $\frac{2k-1}{2}$ each, and thus have total weight of $O(nk)$. We conclude

$$\begin{array}{lllllll}|E_{H_n}| & \geq & |E_{H_{n-1}}| & = & \Omega(\frac{1}{k} \cdot n^{1+\frac{1}{k}}) & = & \Omega(\frac{1}{k} \cdot n^{\frac{1}{k}}) \cdot |E_S| \,. \\ w(E_{H_n}) & \geq & w(E_{H_{n-1}}) & = & \Omega(n^{1+\frac{1}{k}}) & = & \Omega(\frac{1}{k} \cdot n^{\frac{1}{k}}) \cdot w(S) \,.\end{array}$$ ◀

6 Conclusion

We studied online spanners for points in metric spaces. In the Euclidean d-space, we presented an online $(1+\varepsilon)$-spanner algorithm with competitive ratio $O(\varepsilon^{1-d}\log n)$, improving the previous bound of $O_d(\varepsilon^{-(d+1)}\log n)$ from [14]. In fact, the spanner maintained by the algorithm has $O_d(\varepsilon^{1-d}\log\varepsilon^{-1}) \cdot n$ edges, almost matching the (offline) optimal bound of $O_d(\varepsilon^{1-d}) \cdot n$. Moreover, in the plane, a tighter analysis of the same algorithm provides an almost quadratic improvement of the competitive ratio to $O(\varepsilon^{-3/2}\log\varepsilon^{-1}\log n)$, by comparing the online spanner with an instance-optimal spanner directly, circumventing the comparison to an MST (i.e., lightness). Note that, the logarithmic dependence on n is unavoidable due to a $\Omega((\varepsilon^{-1}/\log\varepsilon^{-1})\log n)$ lower bound in the real line [14]. However, our lower bound $\Omega(\varepsilon^{-d})$ under the L_1-norm in $\mathbb{R}^d$ shows a dependence on the dimension. This leads to the following question.

▶ **Question.** *Does the competitive ratio of an online $(1+\varepsilon)$-spanning algorithm for n points in $\mathbb{R}^d$ necessarily grow proportionally with $\varepsilon^{-f(d)} \cdot \log n$, where $\lim_{d\to\infty} f(d) = \infty$?*

Interestingly, for $t \in [(1+\varepsilon)\sqrt{2}, (1-\varepsilon)2]$, we can show that every online t-spanner algorithm in $\mathbb{R}^d$ must have competitive ratio $2^{\Omega(\varepsilon^2 d)}$ (see the full paper for further details).

Next, we studied online spanners in general metrics. We showed that the *ordered greedy* algorithm maintains a spanner with $O(\varepsilon^{-1}\log\varepsilon^{-1}) \cdot n^{1+\frac{1}{k}}$ edges and $O(\varepsilon^{-1}n^{\frac{1}{k}}\log^2 n)$ lightness, with stretch factor $t = (2k-1)(1+\varepsilon)$ for $k \geq 2$ and $\varepsilon \in (0,1)$, for a sequence of n points in a metric space. Moreover, we show that these bounds cannot be significantly improved, by introducing an instance that achieves an $\Omega(\frac{1}{k} \cdot n^{1/k})$ competitive ratio on both sparsity and lightness. Finally, we established the trade-off among stretch, number of edges and lightness for points in ultrametrics, showing that one can maintain a $(2+\varepsilon)$-spanner for ultrametrics with $O(\varepsilon^{-1} \log \varepsilon^{-1}) \cdot n$ edges and $O(\varepsilon^{-2})$ lightness.

References

1 Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms (TALG)*, 2(4):640–660, 2006. doi:10.1145/1198513.1198522.

2 Noga Alon and Yossi Azar. On-line Steiner trees in the Euclidean plane. *Discrete & Computational Geometry*, 10:113–121, 1993. doi:10.1007/BF02573969.

3 Stephen Alstrup, Søren Dahlgaard, Arnold Filtser, Morten Stöckel, and Christian Wulff-Nilsen. Constructing light spanners deterministically in near-linear time. *Theoretical Computer Science*, 2022. doi:10.1016/j.tcs.2022.01.021.

4 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993. doi:10.1007/BF02189308.

5 Baruch Awerbuch, Yossi Azar, and Yair Bartal. On-line generalized Steiner problem. *Theoretical Computer Science*, 324(2-3):313–324, 2004. doi:10.1016/j.tcs.2004.05.021.

6 Surender Baswana. Streaming algorithm for graph spanners—single pass and constant processing time per edge. *Inf. Process. Lett.*, 106(3):110–114, 2008. doi:10.1016/j.ipl.2007.11.001.

7 Surender Baswana, Sumeet Khurana, and Soumojit Sarkar. Fully dynamic randomized algorithms for graph spanners. *ACM Trans. Algorithms*, 8(4):35:1–35:51, 2012. doi:10.1145/2344422.2344425.

8 Ruben Becker, Sebastian Forster, Andreas Karrenbauer, and Christoph Lenzen. Near-optimal approximate shortest paths and transshipment in distributed and streaming models. *SIAM J. Comput.*, 50(3):815–856, 2021. doi:10.1137/19M1286955.

9 Thiago Bergamaschi, Monika Henzinger, Maximilian Probst Gutenberg, Virginia Vassilevska Williams, and Nicole Wein. New techniques and fine-grained hardness for dynamic near-additive spanners. In *Proc. 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1836–1855, 2021. doi:10.1137/1.9781611976465.110.

10 Piotr Berman and Chris Coulston. On-line algorithms for Steiner tree problems. In *Proc. 29th ACM Symposium on Theory of Computing (STOC)*, pages 344–353, 1997. doi:10.1145/258533.258618.

11 Aaron Bernstein, Sebastian Forster, and Monika Henzinger. A deamortization approach for dynamic spanner and dynamic maximal matching. In *Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1899–1918, 2019. doi:10.1137/1.9781611975482.115.

12 Sujoy Bhore and Csaba D. Tóth. Light Euclidean Steiner spanners in the plane. In *Proc. 37th International Symposium on Computational Geometry (SoCG)*, volume 189 of *LIPIcs*, pages 31:1–17. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.SoCG.2021.15.

13 Sujoy Bhore and Csaba D. Tóth. On Euclidean Steiner (1+ε)-spanners. In *Proc. 38th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 187 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.STACS.2021.13.

14 Sujoy Bhore and Csaba D. Tóth. Online Euclidean spanners. In *Proc. 29th European Symposium on Algorithms (ESA)*, volume 204 of *LIPIcs*, pages 116:1–16:19. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.ESA.2021.16.

15 Greg Bodwin, Michael Dinitz, and Yasamin Nazari. Vertex fault-tolerant emulators. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *LIPIcs*, pages 25:1–25:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl. doi:10.4230/LIPIcs.ITCS.2022.25.

16 Greg Bodwin and Sebastian Krinninger. Fully dynamic spanners with worst-case update time. In *Proc. 24th Annual European Symposium on Algorithms (ESA)*, volume 57 of *LIPIcs*, pages 17:1–17:18. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.ESA.2016.17.

17 Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998. See this url.

18 Glencora Borradaile, Hung Le, and Christian Wulff-Nilsen. Greedy spanners are optimal in doubling metrics. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms SODA*, pages 2371–2379, 2019. doi:10.1137/1.9781611975482.145.

19 Prosenjit Bose, Jean-Lou De Carufel, Pat Morin, André van Renssen, and Sander Verdonschot. Towards tight bounds on theta-graphs: More is not always better. *Theor. Comput. Sci.*, 616:70–93, 2016. doi:10.1016/j.tcs.2015.12.017.

20 Prosenjit Bose, Joachim Gudmundsson, and Pat Morin. Ordered theta graphs. *Computational Geometry*, 28(1):11–18, 2004. doi:10.1016/j.comgeo.2004.01.003.

21 Paul B. Callahan and S. Rao Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. 4th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 291–300, 1993. URL: http://dl.acm.org/citation.cfm?id=313559.313777.

22 Shiri Chechik and Christian Wulff-Nilsen. Near-optimal light spanners. *ACM Trans. Algorithms*, 14(3):33:1–33:15, 2018. doi:10.1145/3199607.

23 L. Paul Chew. There are planar graphs almost as good as the complete graph. *J. Comput. Syst. Sci.*, 39(2):205–219, 1989. doi:10.1007/BF01758846.

24 Kenneth L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th ACM Symposium on Theory of Computing (STOC)*, pages 56–65, 1987. doi:10.1145/28395.28402.

25 Michael J. Demmer and Maurice P. Herlihy. The arrow distributed directory protocol. In *Proc. 12th Symposium on Distributed Computing (DISC)*, volume 1499 of *LNCS*, pages 119–133. Springer, 1998. doi:10.1007/BFb0056478.

26 Michael Elkin. Streaming and fully dynamic centralized algorithms for constructing and maintaining sparse spanners. *ACM Trans. Algorithms*, 7(2):20:1–20:17, 2011. doi:10.1145/1921659.1921666.

27 Michael Elkin and Shay Solomon. Fast constructions of lightweight spanners for general graphs. *ACM Trans. Algorithms*, 12(3):29:1–29:21, 2016. doi:10.1145/2836167.

28 P. Erdős. Extremal problems in graph theory. *Theory of Graphs and Its Applications (Proc. Sympos. Smolenice)*, pages 29–36, 1964. see here.

29 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008. doi:10.1137/070683155.

30 Arnold Filtser, Michael Kapralov, and Navid Nouri. Graph spanners by sketching in dynamic streams and the simultaneous communication model. In *Proc. 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1894–1913, 2021. doi:10.1137/1.9781611976465.113.

31 Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. *SIAM J. Comput.*, 49(2):429–447, 2020. doi:10.1137/18M1210678.

32 John Fischer and Sariel Har-Peled. Dynamic well-separated pair decomposition made easy. In *Proc. 17th Canadian Conference on Computational Geometry (CCCG)*, pages 235–238, 2005. see here. URL: http://www.cccg.ca/proceedings/2005/32.pdf.

33 Jie Gao, Leonidas J. Guibas, and An Nguyen. Deformable spanners and applications. *Comput. Geom.*, 35(1-2):2–19, 2006. doi:10.1016/j.comgeo.2005.10.001.

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

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

36 Elena Grigorescu, Young-San Lin, and Kent Quanrud. Online directed spanners and Steiner forests. In *Proc. Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques (APPROX/RANDOM)*, volume 207 of *LIPIcs*, pages 5:1–5:25. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.5.

37 Joachim Gudmundsson and Christian Knauer. Dilation and detours in geometric networks. In *Handbook of Approximation Algorithms and Metaheuristics*, volume 2. Chapman and Hall/CRC, 2nd edition, 2018. see here.

38 Joachim Gudmundsson, Christos Levcopoulos, Giri Narasimhan, and Michiel Smid. Approximate distance oracles for geometric spanners. *ACM Transactions on Algorithms (TALG)*, 4(1):1–34, 2008. doi:10.1145/1328911.1328921.

39 Anupam Gupta, R. Ravi, Kunal Talwar, and Seeun William Umboh. LAST but not least: Online spanners for buy-at-bulk. In Philip N. Klein, editor, *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 589–599, 2017. doi:10.1137/1.9781611974782.38.

40 MohammadTaghi Hajiaghayi, Vahid Liaghat, and Debmalya Panigrahi. Online node-weighted steiner forest and extensions via disk paintings. *SIAM J. Comput.*, 46(3):911–935, 2017. doi:10.1137/14098692X.

41 Sariel Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. AMS, Providence, RI, 2011. see here.

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

43 Maurice Herlihy, Srikanta Tirthapura, and Rogert Wattenhofer. Competitive concurrent distributed queuing. In *Proc. 20th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 127–133, 2001. doi:10.1145/383962.384001.

44 Makoto Imase and Bernard M. Waxman. Dynamic Steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991. doi:10.1137/0404033.

45 J. Mark Keil. Approximating the complete Euclidean graph. In *Proc. 1st Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 318 of *LNCS*, pages 208–213. Springer, 1988. doi:10.1007/3-540-19487-8_23.

46 Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proc 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 798–807, 2004. see here. URL: http://dl.acm.org/citation.cfm?id=982792.982913.

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

48 Hung Le and Shay Solomon. Light Euclidean spanners with Steiner points. In *Proc. 28th European Symposium on Algorithms (ESA)*, volume 173 of *LIPIcs*, pages 67:1–67:22. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.ESA.2020.67.

49 Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Rec.*, 43(1):9–20, 2014. doi:10.1145/2627692.2627694.

50 Joseph Naor, Debmalya Panigrahi, and Mohit Singh. Online node-weighted Steiner tree and related problems. In *Proc. 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 210–219, 2011. doi:10.1109/FOCS.2011.65.

51 Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007. doi:10.1017/CBO9780511546884.

52 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. doi:10.1002/jgt.3190130114.

53 David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989. doi:10.1137/0218050.

54 David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM (JACM)*, 36(3):510–530, 1989. doi:10.1145/65950.65953.

55 Liam Roditty. Fully dynamic geometric spanners. *Algorithmica*, 62(3-4):1073–1087, 2012. doi:10.1007/s00453-011-9504-7.

56 Jim Ruppert and Raimund Seidel. Approximating the d-dimensional complete Euclidean graph. In *Proc. 3rd Canadian Conference on Computational Geometry (CCCG)*, pages 207–210, 1991.

57 Christian Schindelhauer, Klaus Volbert, and Martin Ziegler. Geometric spanners with applications in wireless networks. *Computational Geometry*, 36(3):197–214, 2007. doi:10.1016/j.comgeo.2006.02.001.

58 Michiel H. M. Smid. The well-separated pair decomposition and its applications. In *Handbook of Approximation Algorithms and Metaheuristics*, volume 2. CRC Press, 2nd edition, 2018. URL: https://www.taylorfrancis.com/chapters/edit/10.1201/9781351235426-4/well-separated-pair-decomposition-applications-michiel-smid.

59 Seeun William Umboh. Personal communication, October 2021.

60 Andrew Chi-Chih Yao. Space-time tradeoff for answering range queries (extended abstract). In *Proc. 14th Annual ACM Symposium on Theory of Computing (STOC)*, pages 128–136, 1982. doi:10.1145/800070.802185.