

BALANCING DEGREE, DIAMETER, AND WEIGHT IN EUCLIDEAN SPANNERS*

SHAY SOLOMON[†] AND MICHAEL ELKIN[‡]

Abstract. In a seminal paper from 1995, Arya et al. [*Euclidean spanners: Short, thin, and lanky*, in Proceedings of the 27th Annual ACM Symposium on Theory of Computing, ACM, New York, 1995, pp. 489–498] devised a construction that, for any set S of n points in \mathbb{R}^d and any $\epsilon > 0$, provides a $(1 + \epsilon)$ -spanner with diameter $O(\log n)$, weight $O(\log^2 n) \cdot w(MST(S))$, and constant maximum degree. Another construction from the same work provides a $(1 + \epsilon)$ -spanner with $O(n)$ edges and diameter $O(\alpha(n))$, where α stands for the inverse Ackermann function. There are also a few other known constructions of $(1 + \epsilon)$ -spanners. Das and Narasimhan [*A fast algorithm for constructing sparse Euclidean spanners*, in Proceedings of the 10th Annual ACM Symposium on Computational Geometry (SOCG), ACM, New York, 1994, pp. 132–139] devised a construction with constant maximum degree and weight $O(w(MST(S)))$, but the diameter may be arbitrarily large. In another construction by Arya et al., there is diameter $O(\log n)$ and weight $O(\log n) \cdot w(MST(S))$, but this construction may have arbitrarily large maximum degree. While these constructions address some important practical scenarios, they fail to address situations in which we are prepared to compromise on one of the parameters but cannot afford for this parameter to be arbitrarily large. In this paper we devise a novel *unified* construction that trades gracefully among the maximum degree, diameter, and weight. For a positive integer k our construction provides a $(1 + \epsilon)$ -spanner with maximum degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, weight $O(k \cdot \log_k n \cdot \log n) \cdot w(MST(S))$, and $O(n)$ edges. Note that for $k = O(1)$ this gives rise to maximum degree $O(1)$, diameter $O(\log n)$, and weight $O(\log^2 n) \cdot w(MST(S))$, which is one of the aforementioned results of Arya et al. For $k = n^{1/\alpha(n)}$ this gives rise to diameter $O(\alpha(n))$, weight $O(n^{1/\alpha(n)} \cdot \log n \cdot \alpha(n)) \cdot w(MST(S))$, and maximum degree $O(n^{1/\alpha(n)})$. In the corresponding result from Arya et al., the spanner has the same number of edges and diameter, but its weight and degree may be arbitrarily large. Our bound of $O(\log_k n + \alpha(k))$ on the diameter is optimal under the constraints that the maximum degree is $O(k)$ and the number of edges is $O(n)$. Similarly to the bound of Arya et al., our bound on the weight is optimal up to a factor of $\log n$. Our construction also provides a similar trade-off in the complementary range of parameters, i.e., when the weight should be smaller than $\log^2 n$, but the diameter is allowed to grow beyond $\log n$. Moreover, all our results apply to *doubling metrics*. En route to these results we devise optimal constructions of 1-spanners for general tree metrics, and we employ them to build our Euclidean spanners. Subsequent papers have utilized our constructions of 1-spanners for tree metrics to resolve a long-standing conjecture of Arya et al.

Key words. doubling metrics, Euclidean spaces, Euclidean spanners

AMS subject classifications. 97N70, 97K30, 05C85

DOI. 10.1137/120901295

1. Introduction.

1.1. Euclidean spanners. Consider the weighted complete graph $\mathcal{S} = (S, \binom{S}{2})$ induced by a set S of n points in \mathbb{R}^d , $d \geq 2$. The weight of an edge $(x, y) \in \binom{S}{2}$ for a pair of distinct points $x, y \in S$ is defined to be the Euclidean distance $\|x - y\|$ between

*Received by the editors December 4, 2012; accepted for publication (in revised form) May 2, 2014; published electronically August 14, 2014. A preliminary version of this paper appeared in the Proceedings of ESA'10 [44].

<http://www.siam.org/journals/sidma/28-3/90129.html>

[†]Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel (shay.solomon@weizmann.ac.il). Part of this work was done while this author was a graduate student in the Department of Computer Science, Ben-Gurion University of the Negev, under the support of Clore Fellowship grant 81265410, BSF grant 2008430, and ISF grant 87209011.

[‡]Department of Computer Science, Ben-Gurion University of the Negev, POB 653, Beer-Sheva 84105, Israel (elkinm@cs.bgu.ac.il). This author's work was supported by BSF grant 2008430, by ISF grant 87209011, and by the Lynn and William Frankel Center for Computer Sciences.

x and y . Let $G = (S, E)$ be a spanning subgraph of \mathcal{S} , with $E \subseteq \binom{S}{2}$, and assume that, exactly as in \mathcal{S} , for any edge $e = (x, y) \in E$ its weight $w(e)$ in G is defined to be $\|x - y\|$. For a parameter $\epsilon > 0$, the spanning subgraph G is called a $(1 + \epsilon)$ -spanner for the point set S if for every pair $x, y \in S$ of distinct points the distance¹ $\text{dist}_G(x, y)$ between x and y in G is at most $(1 + \epsilon) \cdot \|x - y\|$. Euclidean spanners were introduced² in 1986 by Chew [16]. Since then they have evolved into an important subarea of computational geometry [33, 19, 40, 34, 3, 18, 20, 4, 21, 5, 39, 1, 10, 22]. (See also the book by Narasimhan and Smid on Euclidean spanners [36], and the references therein.) Also, Euclidean spanners have numerous applications in geometric approximation algorithms [39, 28, 29], geometric distance oracles [28, 30, 29], network design [32, 35], and other areas.

In many of these applications one is required to construct a $(1 + \epsilon)$ -spanner $G = (S, E)$ that satisfies a number of useful properties. First, the spanner should contain $O(n)$ (or nearly $O(n)$) edges. Second, its *weight* $w(G) = \sum_{e \in E} w(e)$ should not be much greater than the weight $w(\text{MST}(S))$ of the minimum spanning tree $\text{MST}(S)$ of S . Third, its *diameter* $\Lambda = \Lambda(G)$ should be small; i.e., for every pair of points $x, y \in S$ there should exist a path P in G that contains at most Λ edges and has weight $w(P) = \sum_{e \in E(P)} w(e) \leq (1 + \epsilon) \cdot \|x - y\|$. Fourth, its *maximum degree* (henceforth, *degree*) $\Delta(G)$ should be small.

In a seminal STOC'95 paper that culminated a long line of research, Arya et al. [4] devised a construction of $(1 + \epsilon)$ -spanners with lightness³ $O(\log^2 n)$, diameter $O(\log n)$, and constant degree. They also devised a construction of $(1 + \epsilon)$ -spanners with diameter $O(\alpha(n))$ (resp., $O(1)$) and $O(n)$ (resp., $O(n \cdot \log^* n)$) edges, where α stands for the inverse Ackermann function [17, 45]. However, in the latter construction the resulting spanners may have arbitrarily large (i.e., $\Omega(n)$) lightness and degree. There are also a few other known constructions of $(1 + \epsilon)$ -spanners. Das and Narasimhan [20] devised a construction with constant degree and lightness, but for which the diameter may be arbitrarily large. (See also [27] for a faster implementation of a spanner construction with constant degree and lightness.) There is also another construction by Arya et al. [4] that guarantees that both the diameter and the lightness are $O(\log n)$, for which the degree may be arbitrarily large. While these constructions address some important practical scenarios, they certainly do not address all of them. In particular, they fail to address situations in which we are prepared to compromise on one of the parameters but cannot afford for this parameter to be arbitrarily large.

In this paper we devise a novel *unified* construction that trades gracefully among the degree, diameter, and weight. For a positive integer k , our construction provides a $(1 + \epsilon)$ -spanner with degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, lightness $O(k \cdot \log_k n \cdot \log n)$, and $O(n)$ edges. Also, we can improve the bound on the diameter from $O(\log_k n + \alpha(k))$ to $O(\log_k n)$, at the expense of increasing the number of edges from $O(n)$ to $O(n \cdot \log^* n)$. Note that for $k = O(1)$ our trade-off gives rise to degree $O(1)$, diameter $O(\log n)$, and lightness $O(\log^2 n)$, which is one of the aforementioned results of [4]. Also, for $k = n^{1/\alpha(n)}$ it gives rise to a spanner with degree $O(n^{1/\alpha(n)})$, diameter $O(\alpha(n))$, and lightness $O(n^{1/\alpha(n)} \cdot \log n \cdot \alpha(n))$. In the corresponding result from [4]

¹In a weighted graph $G = (V, E)$, the *distance* $\text{dist}_G(x, y)$ is the weight of a shortest path in G connecting x and y .

²The term “spanner” was coined by Peleg and Ullman [37], who also introduced spanners for general graphs.

³For convenience, we will henceforth refer to the normalized notion of weight $\Psi(G) = \frac{w(G)}{w(\text{MST}(S))}$, which we call *lightness*.

TABLE 1

A concise comparison of previous and new results. Each column corresponds to a set of parameters that can be achieved simultaneously. For each column the first row indicates whether the result is new or due to [4]. (The first column is due to [4] but can also be achieved from both our trade-offs.) For new results, the second row indicates whether it is obtained by the first (I) or the second (II) trade-off: The first trade-off is degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, and lightness $O(k \cdot \log_k n \cdot \log n)$; the second trade-off is degree $O(1)$, diameter $O(k \cdot \log_k n)$, and lightness $O(\log_k n \cdot \log n)$. The third row indicates the value of k that is substituted into the corresponding trade-off. The next three rows indicate the resulting degree (Δ), diameter (Λ), and lightness (Ψ). The number of edges used in all constructions is $O(n)$. To save space, the O notation is omitted everywhere except for the exponents. The letters δ and ζ stand for arbitrarily small positive constants.

	[4]	[4]	New	New	New	New	New	New
	I,II		I	I	I	II	II	II
k	1		$\log^\delta n$	$2^{\sqrt{\log n}}$	$n^{1/\alpha(n)}$	$\log^\delta n$	$2^{\sqrt{\log n}}$	n^ζ
Δ	1	n	$\log^\delta n$	$2^{\sqrt{\log n}}$	$n^{1/\alpha(n)}$	1	1	1
Λ	$\log n$	$\alpha(n)$	$\frac{\log n}{\log \log n}$	$\sqrt{\log n}$	$\alpha(n)$	$\log^{1+\delta} n$	$2^{O(\sqrt{\log n})}$	n^ζ
Ψ	$\log^2 n$	n	$\log^{2+\delta} n$	$2^{O(\sqrt{\log n})}$	$n^{O(1/\alpha(n))}$	$\frac{\log^2 n}{\log \log n}$	$\log^{3/2} n$	$\log n$

the spanner has the same number of edges and diameter, but its lightness and degree may be arbitrarily large.

In addition, we can achieve lightness $o(\log^2 n)$ at the expense of increasing the diameter. Specifically, for a parameter k the second variant of our construction provides a $(1+\epsilon)$ -spanner with degree $O(1)$, diameter $O(k \cdot \log_k n)$, and lightness $O(\log_k n \cdot \log n)$. For example, for $k = \log^\delta n$, for an arbitrarily small constant $\delta > 0$ we get a $(1+\epsilon)$ -spanner with degree $O(1)$, diameter $O(\log^{1+\delta} n)$, and lightness $O(\frac{\log^2 n}{\log \log n})$.

Our unified construction can be implemented in $O(n \cdot \log n)$ time. This matches the state-of-the-art running time of the aforementioned constructions [4, 27]. See Table 1 for a concise comparison of previous and new results.

To summarize, our construction improves existing constructions for some specific sets of parameters. For other sets of parameters it matches the current state-of-the-art bounds due to [4]. We believe that replacing a number of separate “ad hoc” constructions by a single unified parameterized construction is an important step toward laying a more solid theoretical foundation for the area of Euclidean spanners.

Note that in any construction of spanners with degree $O(k)$, the diameter is $\Omega(\log_k n)$. Also, Chan and Gupta [10] showed that any $(1+\epsilon)$ -spanner with $O(n)$ edges must have diameter $\Omega(\alpha(n))$. Consequently, our upper bound of $O(\log_k n + \alpha(k))$ on the diameter is tight under the constraints that the degree is $O(k)$ and the number of edges is $O(n)$. If we allow $O(n \cdot \log^* n)$ edges in the spanner, then our bound on the diameter is reduced to $O(\log_k n)$, which is again tight under the constraint that the degree is $O(k)$.

In addition, Dinitz, Elkin, and Solomon [22] showed that for any construction of spanners, if the diameter is at most $O(\log_k n)$, then the lightness is at least $\Omega(k \cdot \log_k n)$; conversely, if the lightness is at most $O(\log_k n)$, then the diameter is at least $\Omega(k \cdot \log_k n)$. This lower bound implies that the bound on lightness in both of our trade-offs cannot possibly be improved by more than a factor of $\log n$. The same slack of $\log n$ is present in the result of [4] that guarantees lightness $O(\log^2 n)$, diameter $O(\log n)$, and constant degree.

1.1.1. Spanners for doubling metrics. The *doubling dimension* of a metric (X, δ) is the smallest value ζ such that every ball B in the metric can be covered by

at most 2^ζ balls of half the radius of B . The metric (X, δ) is called *doubling* if its doubling dimension ζ is constant. Spanners for doubling metrics have received much attention in recent years (see, e.g., [11, 31, 10, 26]). In particular, Chan et al. [11] showed that for any doubling metric (X, δ) there exists a $(1 + \epsilon)$ -spanner with constant maximum degree, but this spanner may have arbitrarily large diameter. Chan and Gupta [10] devised a construction of $(1 + \epsilon)$ -spanners for doubling metrics that achieves the optimal trade-off between the number of edges and diameter, but these spanners may have arbitrarily large degree. Gottlieb, Kontorovich, and Krauthgamer [25] devised a construction of $(1 + \epsilon)$ -spanners for doubling metrics with constant degree and logarithmic diameter. We present a single construction of $(1 + \epsilon)$ -spanners for doubling metrics that achieves the optimal trade-off between the degree, diameter, and number of edges in the entire range of parameters. Specifically, for a parameter k our construction provides a $(1 + \epsilon)$ -spanner with maximum degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, and $O(n)$ edges. Also, we can improve the bound on the diameter from $O(\log_k n + \alpha(k))$ to $O(\log_k n)$, at the expense of increasing the number of edges from $O(n)$ to $O(n \cdot \log^* n)$. More generally, we can achieve the same optimal trade-off between the number of edges and diameter as the spanners of [10] do, while also having the optimal maximum degree. In addition, our spanners have lightness $O(k \cdot \log_k n \cdot \log n)$. Also, as in the Euclidean case we obtain a similar trade-off in the complementary range of parameters. Specifically, a variant of our construction provides $(1 + \epsilon)$ -spanners with constant degree, diameter $O(k \cdot \log_k n)$, and lightness $O(\log_k n \cdot \log n)$.

1.2. Spanners for tree metrics. Let ϑ_n be the metric induced by n points v_1, v_2, \dots, v_n lying on the x -axis with coordinates $1, 2, \dots, n$, respectively. In a classical STOC'82 paper [50], Yao showed that there exists a 1-spanner⁴ $G = (V, E)$ for ϑ_n with diameter $O(\alpha(n))$ and $O(n)$ edges, and that this is tight. Chazelle [14] extended the result of [50] to arbitrary tree metrics. Other proofs of Chazelle's result appeared in [2, 9, 49, 41]. We will refer to Yao's construction of 1-spanners for ϑ_n as *list-spanners*, and to Chazelle's (and others') constructions of 1-spanners for tree metrics as *tree-spanners*. Thorup [49] also devised an efficient parallel algorithm for computing this 1-spanner. The problem was also studied for planar metrics [48], general metrics [47], and even for general graphs [8]. (See Chapter 12 in [36] for an excellent survey on this problem.) The problem is also closely related to the extremely well-studied problem of computing partial sums. (See the papers of Tarjan [46], Yao [50], Chazelle and Rosenberg [15], Pătraşcu and Demaine [38], and the references therein.) For a discussion about the relationship between these two problems, see the introduction of [1].

In all constructions [50, 14, 2, 9, 49, 41] of 1-spanners for tree metrics, the degree and lightness of the resulting spanner may be arbitrarily large. Moreover, the constraint that the diameter is $O(\alpha(n))$ implies that the degree must be $n^{\Omega(1/\alpha(n))}$. A similar lower bound on the lightness follows from the result of [22].

In this paper we extend the results of [50, 14, 2, 9, 49, 41] and devise a construction that achieves the *optimal* (up to constant factors) *trade-off between all involved parameters*. Specifically, consider an n -vertex tree T of degree $\Delta(T)$, and let k be a positive integer. Our construction provides a 1-spanner for the tree metric M_T induced by T with degree $O(\Delta(T) + k)$, diameter $O(\log_k n + \alpha(k))$, lightness $O(k \cdot \log_k n)$, and

⁴The graph G is said to be a 1-spanner for ϑ_n if for every pair of distinct vertices $v_i, v_j \in V$ the distance between them in G is equal to their distance $\|i - j\|$ in ϑ_n . Yao stated this problem in terms of partial sums. However, the two statements of the problem are equivalent.

$O(n)$ edges. We can also get a spanner with diameter $O(\log_k n)$, $O(n \cdot \log^* n)$ edges, and the same degree and lightness as above. For the complementary range of diameter, the second variant of our construction provides a 1-spanner with degree $O(\Delta(T))$, diameter $O(k \cdot \log_k n)$, lightness $O(\log_k n)$, and $O(n)$ edges. As was mentioned above, both of these trade-offs are *optimal up to constant factors*.

We show that this general trade-off between various parameters of 1-spanners for tree metrics is useful for deriving new results (and improving existing results) in the context of Euclidean spanners and spanners for doubling metrics. In subsequent papers [23, 12, 43, 13, 42], we as well as other researchers have employed these constructions of 1-spanners for tree metrics to resolve a longstanding conjecture of Arya et al. [4]. Specifically, Arya et al. [4] conjectured that there exists a construction of Euclidean $(1 + \epsilon)$ -spanners with constant degree, logarithmic diameter, and logarithmic lightness. This conjecture was recently resolved in the affirmative by the authors of the current paper in [23]. There we also devised a construction that exhibits a general trade-off between the degree, diameter, and lightness, which is optimal in the entire range of the parameters. Alternative proofs of Arya et al.'s conjecture were provided subsequently in [12, 43, 13, 42]. Remarkably, all the constructions of [23, 12, 43, 13, 42] rely on the constructions of 1-spanners for tree metrics that we devise in the current paper. We also anticipate that our construction of 1-spanners for tree metrics will be found useful in the context of partial sums problems, for constructing transitive-closure spanners (see, e.g., [8, 7, 6]) and for other applications.

1.3. Our and previous techniques. Our starting point is the construction of Arya et al. [4] that achieves diameter $O(\log n)$, lightness $O(\log^2 n)$, and constant degree. The construction of [4] is built in two stages. First, a construction for the 1-dimensional case is devised. Then the 1-dimensional construction is extended to arbitrary constant dimension. For 1-dimensional spaces Arya et al. [4] start by devising a construction of 1-spanners with diameter, lightness, and degree all bounded by $O(\log n)$. This construction is quite simple; it is essentially a flattened version of a deterministic skip-list. Next, by a more involved argument, they show that the degree can be reduced to $O(1)$, at the expense of increasing the stretch parameter from 1 to $1 + \epsilon$. Finally, the generalization of their construction to point sets in the plane (or, more generally, to \mathbb{R}^d) is far more involved. Specifically, to this end Arya et al. [4] employ two main tools. The first is *dumbbell trees*, the theory of which was developed by Arya et al. in the same paper [4]. (See also Chapter 11 of [36].) The second is the bottom-up clustering technique that was developed by Frederickson [24] for topology trees. Roughly speaking, the *dumbbell theorem* of [4] states that for every point set S one can construct a forest \mathcal{D} of $O(1)$ dumbbell trees, in which there exists a tree $T \in \mathcal{D}$ for every pair x, y of points from S , such that the distance $\text{dist}_T(x, y)$ between x and y in T is at most $(1 + \epsilon)$ times their Euclidean distance $\|x - y\|$. Arya et al. employ Frederickson's clustering technique on each of these $O(1)$ dumbbell trees to obtain their ultimate spanner.

Similarly to [4], we start by devising a construction of 1-spanners for the 1-dimensional case. However, our construction achieves both diameter and lightness at most $O(\log n)$, in conjunction with an *optimal degree*⁵ of at most 3. (Note that [4] paid for decreasing the degree from $O(\log n)$ to $O(1)$ by increasing the stretch of the spanner from 1 to $1 + \epsilon$. Our construction achieves stretch 1 in conjunction

⁵Observe that any graph (and, in particular, a 1-spanner) with maximum degree 2 must have diameter at least $\frac{n-1}{2}$.

with logarithmic diameter and lightness, and with the optimal degree 3.) Moreover, our construction is far more general, as it provides the *entire suite* of all possible values of diameter, lightness, and degree, and it is *optimal up to constant factors* in the entire range of parameters. We then proceed to extend it to arbitrary tree metrics. Finally, we employ the dumbbell trees of Arya et al. [4]. Specifically, we construct our 1-spanners for the metrics induced by each of these dumbbell trees and return their union as our ultimate spanner. As a result we obtain a *unified* construction of Euclidean spanners that achieves near-optimal trade-offs in the entire range of parameters. We note that it is unclear whether the construction of Arya et al. [4] can be extended to provide additional combinations between the diameter and lightness other than $O(\log n)$ and $O(\log^2 n)$, respectively; roughly speaking, the logarithms there come from the number of levels in Frederickson's topology trees. In particular, the construction of Arya et al. [4] that achieves diameter $O(\alpha(n))$ and arbitrarily large lightness and degree is based on completely different ideas. On the other hand, our construction yields a stronger result (diameter $O(\alpha(n))$, lightness and degree $n^{O(1/\alpha(n))}$), and this result is obtained by substituting a different parameter into one of our trade-offs. Moreover, our construction is much simpler and more modular than that of [4]. In particular, it does not employ Frederickson's bottom-up clustering technique but rather constructs 1-spanners for dumbbell trees directly.

Also, our construction of 1-spanners for tree metrics (that we use for dumbbell trees) is fundamentally different from the previous constructions due to [50, 14, 2, 9, 49, 41]. In particular, the techniques of [14, 2, 9, 49, 41] for generalizing constructions of 1-spanners from 1-dimensional metrics to general tree metrics ensure that the diameters of the resulting spanners are not (much) greater than the diameter in the 1-dimensional case. However, the degree and/or lightness of spanners for tree metrics that are obtained by these techniques may be arbitrarily large. To overcome this obstacle, we adapt the techniques of [14, 2, 9, 41] to our purposes. Next, we overview this adaptation. A central ingredient in the generalization techniques of [14, 2, 9, 41] is a tree decomposition procedure. Given an n -vertex rooted tree (T, rt) and a parameter k , this procedure computes a set C of $O(k)$ cut vertices. This set has the property that removing all vertices of C from the tree T decomposes T into a collection \mathcal{F} of trees, so that each tree $\tau \in \mathcal{F}$ contains $O(n/k)$ vertices. This decomposition induces a tree $\mathcal{Q} = \mathcal{Q}(\tau, C)$ over the vertex set $C \cup \{rt\}$ in a natural way: a cut vertex $w \in C$ is defined to be a child of its closest ancestor in T that belongs to $C \cup \{rt\}$. For our purposes, it is crucial that the degree of the tree \mathcal{Q} not be (much) greater than the degree of T . In addition, it is essential that each tree $\tau \in \mathcal{F}$ be incident to at most $O(1)$ cut vertices. We devise a novel decomposition procedure that guarantees these two basic properties. Intuitively, our decomposition procedure "slices" the tree in a "path-like" fashion. This path-like nature of our decomposition enables us to keep the degree and lightness of our construction for general tree metrics (essentially) as small as in the 1-dimensional case.

Subsequent work. Since the conference version of this paper was published in ESA'10 [44], a few newer papers have continued this line of research. In [23] the authors of the current paper came up with a construction of spanners which exhibit a better (and, in fact, optimal up to constant factors) trade-off between the degree, diameter, and lightness than the trade-off presented in the current paper. The construction of [23] relies on the current paper in two ways. First, it invokes the optimal construction of 1-dimensional 1-spanners, which we developed in the current paper. Second, the result of [23] is achieved by a transformation that takes

as input a construction of spanners with small degree and diameter but *with no guarantee on the lightness* and produces a construction of spanners with the same (up to constant factors) degree and diameter, and with logarithmic lightness. To achieve its ultimate result, [23] invokes its transformation on the construction of spanners that we developed in the current paper. The transformation of [23] uses the fact that the latter construction (given in the current paper) exhibits an optimal trade-off between the degree and diameter, and it ignores the (suboptimal) bound on the lightness guaranteed by the latter construction.

Following on [23], Chan, Li, and Ning [12] devised an alternative construction of spanners that achieves an optimal combination of degree, diameter, and lightness in a specific (though important) range of these parameters (specifically, for constant degree and logarithmic diameter and lightness). Their constructions extend also to the fault-tolerant setting. Another simpler construction with similar properties was devised by Solomon [43]. A merged version of [12] and [43] was published in the proceedings of ICALP'13 [13]. Yet another related construction (with better properties for the fault-tolerant setting) was devised by Solomon [42]. All these follow-up constructions [12, 43, 13, 42] to [23] use a 1-spanner for tree metrics with constant degree and logarithmic diameter, which we developed in the current paper. We note that the constructions of [12, 43, 13, 42] do not provide a general trade-off between the parameters, while our construction in the current paper (and also the construction in [23]) does.

1.4. Structure of the paper. In section 2 we describe our construction of 1-spanners for tree metrics. Therein we start (section 2.1) by outlining our basic scheme. We proceed (section 2.2) with describing our 1-dimensional construction. In section 2.3 we extend this construction to general tree metrics. Our tree decomposition procedure (which is at the heart of this extension) is described in section 2.3.1. In sections 3 and 4 we derive our results for Euclidean spanners and spanners for doubling metrics, respectively.

1.5. Preliminaries. An n -point metric space $M = (V, dist)$ can be viewed as a weighted complete graph $G(M) = (V, \binom{V}{2}, dist)$ in which for every pair of points $x, y \in V$ the weight of the edge $e = (x, y)$ in $G(M)$ is defined by $w(x, y) = dist(x, y)$. Let G' be a spanning subgraph of M . We say that G' is a t -spanner for M if for every pair $x, y \in V$ of distinct points there exists a path in G' between x and y whose weight (i.e., the sum of all its edge weights) is at most $t \cdot dist(x, y)$. Such a path is called a t -spanner path. The *stretch* of G' is the minimum number t such that G' is a t -spanner for M . Let T be an arbitrary tree, and denote by $V(T)$ the vertex set of T . For any two vertices u, v in T , their (weighted) distance in T is denoted by $dist_T(u, v)$. The tree metric M_T induced by T is defined as $M_T = (V(T), dist_T)$. The *size* of T , denoted by $|T|$, is the number of vertices in T . Finally, for a positive integer n , we denote the set $\{1, 2, \dots, n\}$ by $[n]$.

2. 1-Spanners for tree metrics.

2.1. The basic scheme. Consider an arbitrary n -vertex (weighted) rooted tree (T, rt) , and let M_T be the tree metric induced by T . Clearly, T is both a 1-spanner and a minimum spanning tree (MST) of M_T , but its diameter may be arbitrarily large. We would like to reduce the diameter of this 1-spanner by adding to it some edges. On the other hand, the number of edges of the resulting spanner should still be linear in n . Moreover, the lightness and the maximum degree of the resulting spanner should also be reasonably small.

Let H be a spanning subgraph of M_T . The *monotone distance* between any

two points u and v in H is defined as the minimum number of edges in a 1-spanner path in H that connects them. Two points in M_T are called *comparable* if one is an ancestor of the other in the underlying tree T . The *monotone diameter* (resp., *comparable monotone diameter*) of H , denoted $\Lambda(H)$ (resp., $\bar{\Lambda}(H)$), is defined as the maximum monotone distance in H between any two points (resp., any two comparable points) in M_T . Observe that if any two *comparable* points are connected via a 1-spanner path that consists of at most h edges, then any two *arbitrary* points are connected via a 1-spanner path that consists of at most $2h$ edges. Consequently, $\bar{\Lambda}(H) \leq \Lambda(H) \leq 2 \cdot \bar{\Lambda}(H)$. We henceforth restrict our attention to the comparable monotone diameter in what follows.

Let $k \geq 2$ be a fixed parameter. The first ingredient of the algorithm is to select a set of $O(k)$ *cut vertices* whose removal from T partitions it into a collection of subtrees of size $O(n/k)$ each. (As mentioned in the last paragraph of section 1.3, we also require this set to satisfy several additional properties.) Having selected the cut vertices, the next step of the algorithm is to connect the cut vertices via $O(k)$ edges, so that the monotone distance between any pair of comparable cut vertices will be small. (This phase does not involve a recursive call of the algorithm.) Finally, the algorithm calls itself recursively for each of the subtrees.

We insert all edges of the original tree T into our final spanner H . These edges connect cut vertices and subtrees in the spanner. We note that the spanner contains no other edges that connect cut vertices and subtrees. Moreover, the spanner contains no edges that connect different subtrees.

2.2. 1-Dimensional spaces. In this section we devise an optimal construction of 1-spanners for ϑ_n . (Recall that ϑ_n is the metric induced by n points v_1, v_2, \dots, v_n lying on the x -axis with coordinates $1, 2, \dots, n$, respectively.) Our argument extends easily to any 1-dimensional space.

Denote by P_n the path $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)$ that induces the metric ϑ_n . Note that the edges of P_n (henceforth, *path-edges*) belong to all spanners that we construct.

2.2.1. Selecting the cut vertices. Let $k \geq 2$ be a fixed parameter. The task of selecting the cut vertices in the 1-dimensional case is trivial. (We assume for simplicity that n is an integer power of k .) In addition to the two endpoints v_1 and v_n of the path, we select the $k - 1$ points r_1, r_2, \dots, r_{k-1} to be cut vertices, where for each $i \in [k - 1]$, $r_i = v_{i(n/k)}$. Indeed, by removing the $k + 1$ cut vertices $r_0 = v_1, r_1, \dots, r_{k-1}, r_k = v_n$ from the path (along with their incident edges), we are left with k intervals I_1, I_2, \dots, I_k of length at most n/k each. The two endpoints v_1 and v_n of the path are called the *sentinels*, and they play a special role in the construction. (See Figure 1 for an illustration for the case $k = 2$.)

2.2.2. 1-spanners with low diameter. In this section we devise a construction $H_k(n)$ of 1-spanners for ϑ_n with comparable monotone diameter $\bar{\Lambda}(n) = \bar{\Lambda}(H_k(n))$ in the range $\Omega(\alpha(n)) = \bar{\Lambda}(n) = O(\log n)$. In section 2.2.3 we turn our attention to spanners with larger monotone diameter. Observe that for ϑ_n the notions of comparable monotone diameter and monotone diameter are equivalent.

First, the algorithm connects the $k + 1$ cut vertices $r_0 = v_1, r_1, \dots, r_{k-1}, r_k = v_n$ via one of the aforementioned constructions of 1-spanners from [50, 14, 2, 9, 49, 41] (henceforth, *list-spanner*). (See the first paragraph of section 1.2 for more details.) In other words, $O(k)$ edges between cut vertices are added to the spanner $H_k(n)$ to

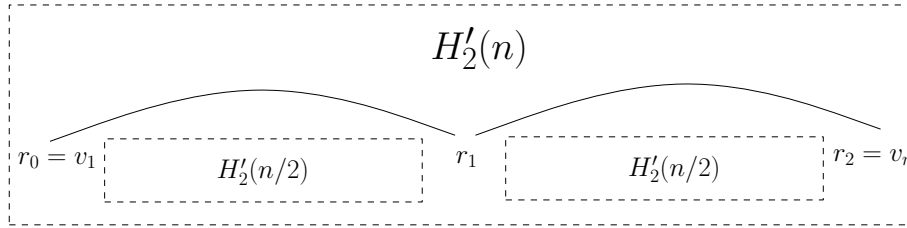


FIG. 1. The construction for $k = 2$. Only the first level of the recursion is illustrated. (Path-edges are not depicted in the figure.) The cut vertex $r_1 = v_{n/2}$ is connected via edges to the two sentinels v_1 and v_n . The construction proceeds recursively for each of the two intervals I_1 and I_2 .

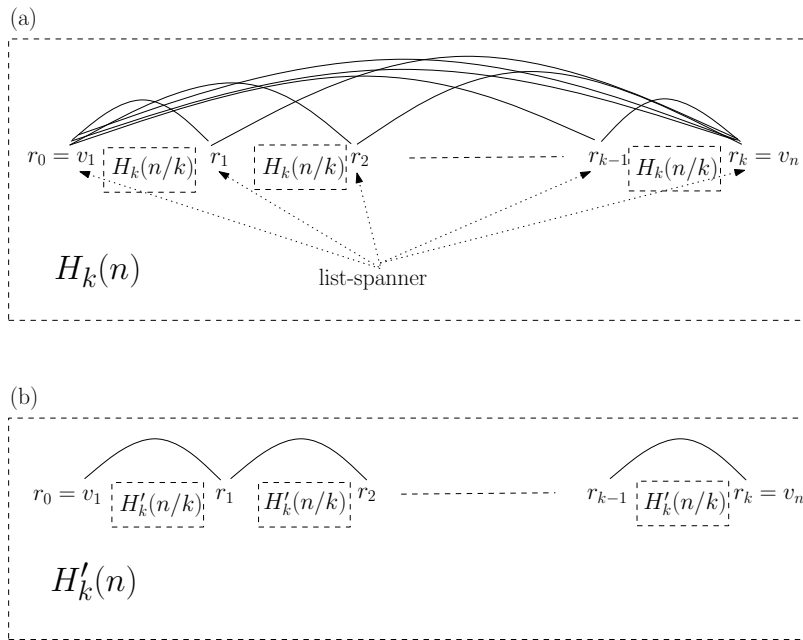


FIG. 2. (a) The construction $H_k(n)$ for a general parameter $k, k \geq 2$. Only the first level of the recursion is illustrated. (Path-edges are not depicted in the figure.) For $H_k(n)$, all the cut vertices are connected via the list-spanner, and, in addition, each of the two sentinels is connected to all other k cut vertices. (b) The construction $H'_k(n)$ for a general parameter $k, k \geq 2$. Each cut vertex r_{i-1} is connected to the next cut vertex r_i in line, $i \in [k]$.

guarantee that the monotone distance in the spanner between any two cut vertices⁶ will be $O(\alpha(k))$. Then the algorithm adds to the spanner $H_k(n)$ edges that connect each of the two sentinels to all other k cut vertices. Finally, the algorithm calls itself recursively for each of the intervals I_1, I_2, \dots, I_k . At the bottom level of the recursion, i.e., when $n \leq k$, the algorithm uses the list-spanner to connect all points, and, in addition, it adds to the spanner edges that connect each of the two sentinels v_1 and v_n to all the other $n - 1$ points. (See Figure 2(a) for an illustration.)

We next analyze the properties of the constructed spanner $H_k(n)$.

LEMMA 2.1. *The number of edges in $H_k(n)$ is at most $O(n)$.*

Proof. Denote by $E(n)$ the number of edges in $H_k(n)$, excluding edges of P_n .

⁶In the 1-dimensional case any two points are comparable.

Clearly, $E(n)$ satisfies the recurrence $E(n) \leq O(k) + k \cdot E(n/k)$, with the base condition $E(q) = O(q)$ for all $q \leq k$, yielding $E(n) = O(n)$. Including edges of P_n , the number of edges increases by $n - 1$ units. \square

LEMMA 2.2. *The maximum degree $\Delta(H_k(n))$ of $H_k(n)$ is at most $k + 2$.*

Proof. Denote by $\Delta(n)$ the maximum degree of a vertex in $H_k(n)$, excluding edges of P_n . Clearly, $\Delta(n)$ satisfies the recurrence $\Delta(n) \leq \max\{k, \Delta(n/k)\}$, with the base condition $\Delta(q) \leq q - 1$ for all $q \leq k$, yielding $\Delta(n) \leq k$. Including edges of P_n , the maximum degree increases by at most two units. \square

LEMMA 2.3. *The lightness $\Psi(H_k(n))$ of $H_k(n)$ is at most $O(k \cdot \log_k n)$.*

Proof. Denote by $w(n)$ the weight of $H_k(n)$, excluding edges of P_n . Note that at most $O(k)$ edges are added between cut vertices. Each of these edges has weight at most $n - 1$. The total weight of all edges within an interval I_i is at most $w(n/k)$. Hence $w(n)$ satisfies the recurrence $w(n) \leq O(n \cdot k) + k \cdot w(n/k)$, with the base condition $w(q) = O(q^2)$ for all $q \leq k$. It follows that $w(n) = O(n \cdot k \cdot \log_k n) = O(k \cdot \log_k n) \cdot w(MST(\vartheta_n))$. Including edges of P_n , the weight increases by $w(P_n) = n - 1$ units. \square

LEMMA 2.4. *The comparable monotone diameter $\bar{\Lambda}(n) = \bar{\Lambda}(H_k(n))$ of $H_k(n)$ is at most $O(\log_k n + \alpha(k))$.*

Proof. The *monotone radius* $R(n)$ of $H_k(n)$ is defined as the maximum monotone distance in $H_k(n)$ between one of the sentinels (either v_1 or v_n) and some other point in ϑ_n . Let v_j be a point in ϑ_n , and let i be the index such that $v_j \in \{r_i\} \cup I_{i+1}$. (In other words, i is the index such that $i(n/k) \leq j < (i + 1)(n/k)$.) We construct a 1-spanner path $\Pi(v_1, v_j)$ in $H_k(n)$ connecting the sentinel v_1 and the point v_j recursively. If $j = i(n/k)$, then v_j is the cut vertex r_i ; in this case the path $\Pi(v_1, v_j)$ consists of the single edge (v_1, v_j) . Otherwise, $j > i(n/k)$ and $v_j \in I_{i+1}$. In this case the path $\Pi(v_1, v_j)$ starts with the two edges $(v_1, v_{i(n/k)}) = (v_1, r_i)$, $(v_{i(n/k)}) = (r_i, v_{i(n/k)+1})$. The point $v_{i(n/k)+1}$ is a sentinel of the $(i + 1)$ th interval I_{i+1} . Hence, the path $\Pi(v_1, v_j)$ will continue recursively, from $v_{i(n/k)+1}$ to v_j . (See Figure 3 for an illustration.) It follows that the monotone radius $R(n)$ satisfies the recurrence $R(n) \leq 2 + R(n/k)$, with the base condition $R(q) = 1$ for all $q \leq k$, yielding $R(n) = O(\log_k n)$.

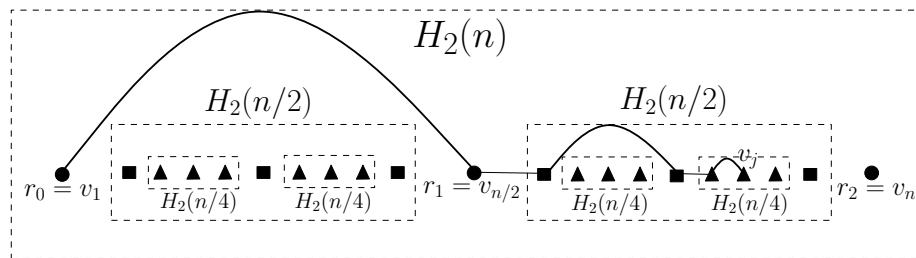


FIG. 3. *An illustration of the path $\Pi(v_1, v_j)$ between v_1 and v_j in $H_k(n)$ for the case $k = 2$. The cut vertices of the first (resp., second; third) level of the recursion are depicted by filled circles (resp., squares; triangles). We assume that v_j is the second triangle from the right; i.e., it is a cut vertex of the third level of the recursion, which belongs to interval I_2 . The first two edges in $\Pi(v_1, v_j)$ are $(v_1, r_1 = v_{n/2}), (v_{n/2}, v_{n/2+1})$, where $v_{n/2+1}$ is the left sentinel of the interval I_2 . Consider the partition of the points in $I' = I_2$ into intervals I'_1 and I'_2 . Note that v_j belongs to interval I'_2 . The next two edges in $\Pi(v_1, v_j)$ lead us to the left sentinel of I'_2 (i.e., the third triangle from the right). The last edge of $\Pi(v_1, v_j)$ connects the left sentinel of I'_2 to v_j .*

Having bounded the monotone radius $R(n)$ of $H_k(n)$, we turn to bounding the comparable monotone diameter $\bar{\Lambda}(n)$ of $H_k(n)$. Note that the bound $R(n) = O(\log_k n)$ on the monotone radius does not imply a bound of $O(\log_k n)$ for $\bar{\Lambda}(n)$. The reason

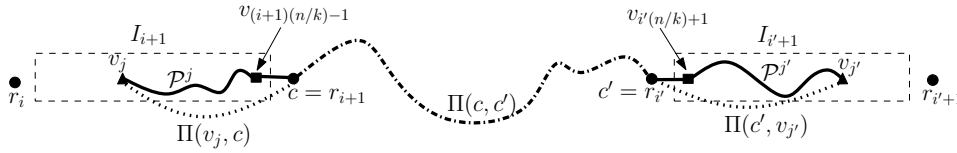


FIG. 4. An illustration of the path $\Pi(v_j, v_{j'})$ between v_j and $v_{j'}$ in $H_k(n)$ (in the case that $v_j \in I_{i+1}$ and $v_{j'} \in I_{i'+1}$), which is given as the concatenation of the three 1-spanner paths $\Pi(v_j, c)$, $\Pi(c, c')$, and $\Pi(c', v_{j'})$, where (i) $\Pi(v_j, c)$ is depicted as a dotted line, and is given as the union of path \mathcal{P}^j (depicted as a solid curved line) and edge $(v_{(i+1)(n/k)-1}, v_{(i+1)(n/k)} = c = r_{i+1})$ (depicted as a solid line), (ii) $\Pi(c, c')$ is depicted as a dash-dotted line, and (iii) $\Pi(c', v_{j'})$ is depicted as a dotted line, and is given as the union of edge $(v_{i'(n/k)} = c' = r_{i'}, v_{i'(n/k)+1})$ (depicted as a solid line) and path $\mathcal{P}^{j'}$ (depicted as a solid curved line).

for this is that a path connecting a pair v_i, v_j of vertices through a sentinel is not monotone.

Consider an arbitrary pair $v_j, v_{j'}$ of points in ϑ_n , with $j < j'$, and let i, i' be the pair of indices such that $v_j \in \{r_i\} \cup I_{i+1}$, $v_{j'} \in \{r_{i'}\} \cup I_{i'+1}$. (In other words, i and i' are the indices such that $i(n/k) \leq j < (i+1)(n/k)$, $i'(n/k) \leq j' < (i'+1)(n/k)$.) Suppose first that v_j and $v_{j'}$ belong to the same interval $I_{i+1} = I_{i'+1}$. In this case it can be argued inductively that the monotone distance between v_j and $v_{j'}$ is at most $\max\{\bar{\Lambda}(n/k), R(n/k) + 1\}$. (The monotone distance between v_j and $v_{j'}$ is at most $R(n/k) + 1$ in the case that $v_j = r_i$.) Otherwise, we construct a 1-spanner path $\Pi(v_j, v_{j'})$ in $H_k(n)$ connecting v_j and $v_{j'}$ that consists of at most $O(\alpha(k)) + 2R(n/k)$ edges. (See Figure 4 for an illustration.) Let $P = (v_j, v_{j+1}, \dots, v_{j'})$ be the subpath of P_n between v_j and $v_{j'}$, and let c and c' denote the first and last cut vertices along P , respectively. The path $\Pi(v_j, v_{j'})$ starts with a 1-spanner path $\Pi(v_j, c)$ between v_j and c . If $j = i(n/k)$, then v_j is the cut vertex r_i , and so $c = v_j = r_i$; in this case $\Pi(v_j, c)$ is defined as the empty path. Otherwise $j > i(n/k)$ and $v_j \in I_{i+1}$, and so $c = r_{i+1}$. By the recursive nature of the construction, $H_k(n)$ contains as a subgraph a 1-spanner for the $(i+1)$ th interval I_{i+1} , having monotone radius at most $R(n/k)$; hence, there is a 1-spanner path \mathcal{P}^j between v_j and the (right) sentinel $v_{(i+1)(n/k)-1}$ of I_{i+1} that consists of at most $R(n/k)$ edges. The path $\Pi(v_j, c)$ is defined as the concatenation of the path \mathcal{P}^j and the edge $(v_{(i+1)(n/k)-1}, v_{(i+1)(n/k)} = c = r_{i+1})$; note that $\Pi(v_j, c)$ consists of at most $R(n/k) + 1$ edges. The path $\Pi(v_j, v_{j'})$ continues with a 1-spanner path $\Pi(c, c')$ between the two cut vertices c and c' that consists of at most $O(\alpha(k))$ edges; the existence of such a path is guaranteed by the list-spanner. The path $\Pi(v_j, v_{j'})$ finishes with a 1-spanner path $\Pi(c', v_{j'})$ between c' and $v_{j'}$. Notice that $c' = r_{i'}$. If $j' = i'(n/k)$, then $v_{j'}$ is the cut vertex $r_{i'}$, and so $c' = r_{i'} = v_{j'}$; in this case $\Pi(c', v_{j'})$ is defined as the empty path. Otherwise $j' > i'(n/k)$ and $v_{j'} \in I_{i'+1}$, and so $c' = r_{i'} \neq v_{j'}$. By the recursive nature of the construction, $H_k(n)$ contains as a subgraph a 1-spanner for the $(i'+1)$ th interval $I_{i'+1}$, having monotone radius at most $R(n/k)$; hence, there is a 1-spanner path $\mathcal{P}^{j'}$ between the (left) sentinel $v_{i'(n/k)+1}$ of $I_{i'+1}$ and $v_{j'}$ that consists of at most $R(n/k)$ edges. The path $\Pi(c', v_{j'})$ is defined as the concatenation of the edge $(v_{i'(n/k)} = c' = r_{i'}, v_{i'(n/k)+1})$ and the path $\mathcal{P}^{j'}$; note that $\Pi(c', v_{j'})$ consists of at most $R(n/k) + 1$ edges. Thus, $\Pi(v_j, v_{j'})$ is given as the concatenation of the three 1-spanner paths $\Pi(v_j, c)$, $\Pi(c, c')$, and $\Pi(c', v_{j'})$; i.e., $\Pi(v_j, v_{j'}) = \Pi(v_j, c) \circ \Pi(c, c') \circ \Pi(c', v_{j'})$. It is easy to see that $\Pi(v_j, v_{j'})$ is a 1-spanner path in $H_k(n)$ between v_j and $v_{j'}$ that consists of at most $O(\alpha(k)) + 2R(n/k)$ edges. It follows that $\bar{\Lambda}(n)$ satisfies the recurrence $\bar{\Lambda}(n) \leq \max\{\bar{\Lambda}(n/k), O(\alpha(k)) + 2R(n/k)\}$, with the base condition $\bar{\Lambda}(q) = O(\alpha(q))$

for all $q \leq k$. Hence $\bar{\Lambda}(n) = O(\log_k n + \alpha(k))$. \square

LEMMA 2.5. *The worst-case running time of the algorithm is at most $O(n)$.*

Proof. Denote the worst-case running time of the algorithm by $t(n)$, excluding the time needed to add the edges of P_n to the spanner. The list-spanner of [50, 14, 2, 9, 49, 41] can be implemented in linear time. By construction, $t(n)$ satisfies the recurrence $t(n) \leq O(k) + k \cdot t(n/k)$, with the base condition $t(q) = O(q)$ for all $q \leq k$, yielding $t(n) = O(n)$. Adding the edges of P_n to the spanner takes another $O(n)$ time. Hence, the overall running time of the algorithm is $O(n)$. \square

Finally, we present a simple transformation of this construction that reduces the maximum degree from $k + 2$ to $k + 1$, without increasing any of the other parameters by more than a constant factor.

Transformation. Notice that the degree of the two sentinels $r_0 = v_1$ and $r_k = v_n$ in $H_k(n)$ is at most $k + 1$. Indeed, each of the two sentinels may be incident to all other k cut vertices; in addition, v_1 (resp., v_n) is incident to its right neighbor v_2 (resp., left neighbor v_{n-1}) along the path P_n . On the other hand, a nonsentinel cut vertex $r_i = v_{i(n/k)}$, $i \in [k - 1]$, is incident to both its left neighbor $v_{i(n/k)-1}$ and its right neighbor $v_{i(n/k)+1}$ along P_n ; therefore, its degree may be as large as $k + 2$. To decrease the degree of a nonsentinel cut vertex r_i , $i \in [k - 1]$, we replace it by a pair of consecutive (along P_n) cut vertices r_i^L, r_i^R , where $r_i^L = r_i = v_{i(n/k)}$, $r_i^R = v_{i(n/k)+1}$. The two sentinels v_1 and v_n remain unchanged; for technical convenience, we define $r_0^L = r_0^R = r_0 = v_1$, $r_k^L = r_k^R = r_k = v_n$. Thus, instead of having $k + 1$ cut vertices, we will now have $2k$ cut vertices, namely, $r_0^L = r_0^R = v_1$, $r_1^L = v_{(n/k)}$, $r_1^R = v_{(n/k)+1}, \dots, r_{k-1}^L = v_{(k-1)(n/k)}$, $r_{k-1}^R = v_{(k-1)(n/k)+1}$, $r_k^L = r_k^R = v_n$. By removing these $2k$ cut vertices from P_n (along with their incident edges), we are left with k intervals $\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_k$ of length at most $n/k - 1$ each, where $\tilde{I}_i = I_i \setminus \{r_{i-1}^R\}$ for each index $i \in [k]$. We transform $H_k(n)$ by replacing each edge (r_i, r_j) in it, $0 \leq i < j \leq k$, with the edge (r_i^R, r_j^L) ; this transformation is applied recursively for each of the intervals $\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_k$ and is repeated as long as the number of points in an interval is greater than k . We denote the resulting graph by $G_k(n)$.

Analysis of the transformation. Fix an arbitrary index $\ell \in [k - 1]$. Clearly, the number of all edges (r_i, r_ℓ) in $H_k(n)$ such that $0 \leq i < \ell$ is no greater than ℓ ; also, the number of all edges (r_ℓ, r_j) in $H_k(n)$ such that $\ell < j \leq k$ is no greater than $k - \ell$. Thus, by construction, the number of cut vertices that are incident to r_ℓ^L (resp., r_ℓ^R) in $G_k(n)$ is at most $\ell \leq k - 1$ (resp., at most $k - \ell \leq k - 1$). Including edges of P_n , the degree of both r_ℓ^L and r_ℓ^R is at most $k + 1$, for each index $\ell \in [k - 1]$. Also, notice that the degree of the two sentinels v_1 and v_n remains as in the basic construction (before the transformation), and so it is at most $k + 1$ as well. Therefore, after the first level of recursion, each of the $2k$ cut vertices $r_0^L = r_0^R, r_1^L, r_1^R, \dots, r_{k-1}^L, r_{k-1}^R, r_k^L = r_k^R$ has degree at most $k + 1$ including edges of P_n . Note also that, excluding edges of P_n , all other vertices have degree zero. In subsequent levels of recursion, the degree of these $2k$ cut vertices remains unchanged, whereas the degree of the other vertices may increase. By employing this argument inductively, we conclude that the maximum degree of $G_k(n)$ is at most $k + 1$, as required. It is easy to see that, similarly to $H_k(n)$, the graph $G_k(n)$ is a 1-spanner for ϑ_n ; moreover, it has the same (up to constant factors) diameter, lightness, and number of edges as those of $H_k(n)$.

Summarizing, we have proved the following theorem.

THEOREM 2.6. *For any n -point 1-dimensional space and a parameter $k \geq 2$ there exists a 1-spanner with maximum degree at most $k + 1$, diameter $O(\log_k n + \alpha(k))$, lightness $O(k \cdot \log_k n)$, and with $O(n)$ edges. The running time of this construction is $O(n)$.*

Remark. For $k = 2$, we get the optimal degree 3, in conjunction with logarithmic diameter and lightness. The same result also follows from Theorem 2.11 below.

2.2.3. 1-spanners with high diameter. In this section we devise a construction $H'_k(n)$ of 1-spanners for ϑ_n with comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}(H'_k(n))$ in the range $\bar{\Lambda}'(n) = \Omega(\log n)$.

The algorithm connects the $k + 1$ cut vertices $r_0 = v_1, r_1, \dots, r_{k-1}, r_k = v_n$ via a path of length k ; i.e., it adds the edges $(r_0, r_1), (r_1, r_2), \dots, (r_{k-1}, r_k)$ into the spanner. In addition, it calls itself recursively for each of the intervals I_1, I_2, \dots, I_k . At the bottom level of the recursion, i.e., when $n \leq k$, the algorithm adds no additional edges to the spanner. (See Figures 1 and 2(b) for an illustration.)

We next analyze the properties of the constructed spanner $H'_k(n)$.

LEMMA 2.7. *The maximum degree $\Delta(H'_k(n))$ of $H'_k(n)$ is at most 4. (Thus the number of edges is at most $2n$.)*

Proof. Denote by $\Delta'(n)$ the maximum degree of a vertex in $H'_k(n)$, excluding edges of P_n . Clearly, $\Delta'(n)$ satisfies the recurrence $\Delta'(n) \leq \max\{2, \Delta'(n/k)\}$, with the base condition $\Delta'(q) = 0$ for all $q \leq k$, yielding $\Delta'(n) \leq 2$. Including edges of P_n , the maximum degree increases by at most two units. \square

LEMMA 2.8. *The lightness $\Psi(H'_k(n))$ of $H'_k(n)$ is at most $O(\log_k n)$.*

Proof. Denote by $w'(n)$ the weight of $H'_k(n)$, excluding edges of P_n . Note that the weight of the path connecting all $k + 1$ cut vertices is equal to $n - 1$. The total weight of all edges within an interval I_i is at most $w'(n/k)$. Hence $w'(n)$ satisfies the recurrence $w'(n) \leq n - 1 + k \cdot w'(n/k)$, with the base condition $w'(q) \leq q - 1$ for all $q \leq k$. It follows that $w'(n) = O(n \cdot \log_k n) = O(\log_k n) \cdot w(MST(\vartheta_n))$. Including edges of P_n , the weight increases by $w(P_n) = n - 1$ units. \square

LEMMA 2.9. *The comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}(H'_k(n))$ of $H'_k(n)$ is at most $O(k \cdot \log_k n)$.*

Proof. It is easy to see that the monotone radius $R'(n)$ of $H'_k(n)$, defined as the maximum monotone distance in $H'_k(n)$ between one of the sentinels and some other point in ϑ_n , satisfies the recurrence $R'(n) \leq k + R'(n/k)$, with the base condition $R'(q) \leq q - 1$ for all $q \leq k$. Hence, $R'(n) = O(k \cdot \log_k n)$. Using reasoning similar to that of Lemma 2.4 from section 2.2.2, we get that the comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}(H'_k(n))$ of $H'_k(n)$ satisfies the recurrence $\bar{\Lambda}'(n) \leq \max\{\bar{\Lambda}'(n/k), k + 2R'(n/k)\}$, with the base condition $\bar{\Lambda}'(q) \leq q - 1$ for all $q \leq k$. It follows that $\bar{\Lambda}'(n) = O(k \cdot \log_k n)$. \square

LEMMA 2.10. *The worst-case running time of the algorithm is at most $O(n)$.*

Proof. Denote the worst-case running time of the algorithm by $t'(n)$, excluding the time needed to add the edges of P_n to the spanner. It is easy to see that $t'(n)$ satisfies the recurrence $t'(n) \leq O(k) + k \cdot t'(n/k)$, with the base condition $t'(q) = O(1)$ for all $q \leq k$, yielding $t'(n) = O(n)$. Adding the edges of P_n to the spanner takes another $O(n)$ time. Hence, the overall running time of the algorithm is $O(n)$. \square

We remark that the spanner $H'_k(n)$ is a planar graph.

Finally, by applying a degree-reducing transformation similar to that presented in section 2.2.2, we can reduce the maximum degree of this construction from 4 to 3 without increasing any of the other parameters by more than a constant factor. Summarizing, we have proved the following theorem.

THEOREM 2.11. *For any n -point 1-dimensional space and a parameter k , there exists a 1-spanner with maximum degree 3, diameter $O(k \cdot \log_k n)$, and lightness $O(\log_k n)$. Moreover, this 1-spanner is a planar graph. The running time of this construction is $O(n)$.*

2.3. General tree metrics. In this section we extend the constructions of section 2.2 from line metrics to general tree metrics.

2.3.1. Selecting the cut vertices. In this section we present a procedure for selecting, given a tree T , a subset of $O(k)$ vertices whose removal from the tree partitions it into subtrees of size $O(|T|/k)$ each. This subset will also satisfy several additional useful properties.

Let (T, rt) be a rooted tree. For an inner vertex v in T with $ch(v)$ children, we denote its children, from left to right, by $c_1(v), c_2(v), \dots, c_{ch(v)}(v)$. Suppose without loss of generality that the size of the subtree $T_{c_1(v)}$ of v is no smaller than the size of any other subtree of v ; i.e., $|T_{c_1(v)}| \geq |T_{c_2(v)}|, |T_{c_3(v)}|, \dots, |T_{c_{ch(v)}(v)}|$. (This assumption can be guaranteed by a straightforward procedure that runs in linear time.) We say that the vertex $c_1(v)$ is the *left-most* child of v . Also, an edge in T is called *left-most* if it connects a vertex v in T and its left-most child $c_1(v)$. We denote by $P(v) = (v, c_1(v), \dots, l(v))$ the path of left-most edges leading down from v to some leaf $l(v)$ in the subtree T_v of T rooted at v ; the leaf $l(v)$ is referred to as the *left-most vertex* in T_v . Also, let $l(T) = l(rt)$ denote the left-most vertex in the entire tree T . An inner vertex v in T is called *d-balanced* for $d \geq 1$, or simply *balanced* if d is clear from the context, if $|T_{c_1(v)}| \leq |T| - d$. The first (i.e., closest to v) balanced vertex along $P(v)$ is denoted by $b(v)$; if no vertex along $P(v)$ is balanced, we write $b(v) = NULL$. Observe that for $|T| \geq 2d$ we have $|T| - d \geq d \geq 1$; in this case the next-to-last vertex along $P(v)$ (namely, the parent $\pi(l(v))$ of $l(v)$ in T) is balanced. Hence, in this case $b(v) \neq NULL$.

Next, we present the procedure *CV* (standing for cut vertices), which accepts as input a rooted tree (T, rt) and a parameter $d \geq 1$ and returns as output a subset of $V(T)$. If $|T| < 2d$, the procedure returns the empty set \emptyset . Otherwise $|T| \geq 2d$, and so the first balanced vertex $b = b(rt)$ along $P(rt)$ satisfies $b \neq NULL$. In this case for each child $c_i(b)$ of b , $i \in [ch(b)]$, the procedure recursively constructs the subset $C_i = CV((T_{c_i(b)}, c_i(b)), d)$ and then returns as output the vertex set $\bigcup_{i=1}^{ch(b)} C_i \cup \{b\}$. (See Figure 5 for an illustration.)

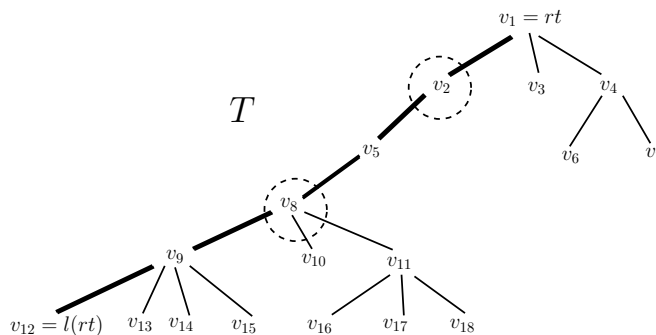


FIG. 5. A rooted tree (T, rt) with $n = |T| = 18$ vertices $v_1 = rt, v_2, \dots, v_{18}$. The edges of $P(rt)$ are depicted by bold lines. The first 6-balanced vertex along $P(rt)$ is v_2 . The procedure *CV* on input (T, rt) and $d = 6$ returns the subset $\{v_2, v_8\}$.

It is easy to see that the running time of this procedure is linear in $|T|$.

Let (T, rt) be an n -vertex rooted tree, and let $d \geq 1$ be a fixed parameter. For convenience, we define $n_i = |T_{c_i(b)}|$ for each $i \in [ch(b)]$. Next, we analyze the properties of the set $C = CV((T, rt), d)$ of *cut vertices*.

Observe that for $n < 2d$, $C = \emptyset$, and for $n \geq 2d$, C is nonempty.

Next, we provide an upper bound on $|C|$ in the case $n \geq 2d$.

LEMMA 2.12. For $n \geq 2d$, $|C| \leq (n/d) - 1$.

Proof. The proof is by induction on $n = |T|$.

Basis: $2d \leq n < 3d$. Fix an index $i \in [ch(b)]$. Since b is balanced, we have

$$n_i \leq n_1 \leq n - d < 2d,$$

implying that $C_i = \emptyset$. It follows that $C = \bigcup_{i=1}^{ch(b)} C_i \cup \{b\} = \{b\}$, and so $|C| = 1 \leq (n/d) - 1$.

Induction step: We assume the correctness of the statement for all smaller values of n , $n \geq 3d$, and prove it for n . Let I be the set of all indices i in $[ch(b)]$ for which $n_i \geq 2d$. Observe that for each $i \in [ch(b)] \setminus I$, $C_i = \emptyset$, and by the induction hypothesis, for each $i \in I$, $|C_i| \leq (n_i/d) - 1$. By construction, the vertex sets $C_1, C_2, \dots, C_{ch(b)}$ and $\{b\}$ are pairwise disjoint, and $C = \bigcup_{i=1}^{ch(b)} C_i \cup \{b\}$. Hence

$$(2.1) \quad |C| = \sum_{i=1}^{ch(b)} |C_i| + 1 = \sum_{i \in I} |C_i| + 1 \leq \sum_{i \in I} ((n_i/d) - 1) + 1.$$

The analysis splits into three cases depending on the value of $|I|$.

Case 1: $|I| = 0$. Equation (2.1) yields $|C| \leq 1 \leq (n/d) - 1$.

Case 2: $|I| = 1$. Since $n_1 = \max\{n_i \mid i \in [ch(b)]\}$, we conclude that the only index i such that $n_i \geq 2d$ in this case is $i = 1$. Hence $I = \{1\}$. Since b is balanced, $n_1 \leq n - d$, and so (2.1) yields

$$|C| \leq (n_1/d) - 1 + 1 \leq (n - d)/d = (n/d) - 1.$$

Case 3: $|I| \geq 2$. Clearly, $\sum_{i \in I} n_i \leq n - 1$, and so (2.1) yields

$$|C| \leq \sum_{i \in I} ((n_i/d) - 1) + 1 = \sum_{i \in I} (n_i/d) - |I| + 1 \leq (n - 1)/d - 2 + 1 \leq (n/d) - 1. \quad \square$$

Let $b = b(rt)$, and let \overline{T}_b be the subtree of T obtained by removing the subtree T_b from T . We use the following claim to prove Lemma 2.14.

CLAIM 2.13. $|\overline{T}_b| < d$.

Proof. If $b = rt$, then \overline{T}_b is empty and the assertion of the claim is immediate. Otherwise, consider the parent $\pi(b)$ of b in T . Since b is the first (i.e., closest to rt) balanced vertex along $P(rt)$, $\pi(b)$ is nonbalanced, and so $|T_b| = |T_{c_1(\pi(b))}| > n - d$. Hence $|\overline{T}_b| = n - |T_b| < d$, and we are done. \square

For a subset U of $V(T)$, we denote by $T \setminus U$ the forest obtained from T by removing all vertices in U along with the edges that are incident to them.

LEMMA 2.14. The size of any subtree in the forest $T \setminus C$ is smaller than $2d$.

Proof. The proof is by induction on $n = |T|$. The basis $n < 2d$ is trivial.

Induction step: We assume the correctness of the statement for all smaller values of n , $n \geq 2d$, and prove it for n . First, note that $b = b(rt) \in C$. Also, observe that for $n \geq 2d$,

$$(2.2) \quad T \setminus C = \bigcup_{i=1}^{ch(b)} (T_{c_i(b)} \setminus C_i) \cup \{\overline{T}_b\}.$$

Consider a subtree T' in the forest $T \setminus C$. By (2.2), either $T' = \overline{T}_b$, or T' belongs to the forest $T_{c_i(b)} \setminus C_i$ for some index $i \in [ch(b)]$. In the former case, the size bound

follows from Claim 2.13, whereas in the latter case it follows from the induction hypothesis. \square

Any subset U of $V(T)$ induces a forest $\mathcal{Q}(T, U)$ over U in the natural way: a vertex $v \in U$ is defined to be a child of its closest ancestor in T that belongs to U . Define $\mathcal{Q} = \mathcal{Q}(T, C)$. Observe that for $n < 2d$, $C = \emptyset$, and so $\mathcal{Q} = \emptyset$. Also, for $n \geq 2d$, C is nonempty and $b = b(rt) \neq \text{NULL}$.

LEMMA 2.15. *For $n \geq 2d$, \mathcal{Q} is a spanning tree of C rooted at $b = b(rt)$ such that, for each vertex v in C , the number of children of v in \mathcal{Q} , denoted $ch_{\mathcal{Q}}(v)$, is no greater than the corresponding number $ch(v)$ in T .*

Remark. This lemma implies that $\Delta(\mathcal{Q}) \leq \Delta(T)$.

Proof. The proof is by induction on $n = |T|$.

Basis: $2d \leq n < 3d$. In this case $C = \{b\}$, and so \mathcal{Q} consists of a single root vertex b .

Induction step: We assume the correctness of the statement for all smaller values of n , $n \geq 3d$, and prove it for n . Let I be the set of all indices i in $[ch(b)]$ for which $n_i \geq 2d$, and write $I = \{i_1, i_2, \dots, i_{|I|}\}$. Observe that for each index $i \in [ch(b)] \setminus I$, $C_i = \emptyset$, and so $\mathcal{Q}(T_{C_i(b)}, C_i)$ is an empty tree. By the induction hypothesis, for each $i \in I$, $\mathcal{Q}_i = \mathcal{Q}(T_{C_i(b)}, C_i)$ is a spanning tree of C_i rooted at $b_i = b(C_i(b)) \neq \text{NULL}$ in which the number of children of each vertex is no greater than the corresponding number in $T_{C_i(b)}$. By definition, the only children of b in \mathcal{Q} are the roots $b_{i_1}, b_{i_2}, \dots, b_{i_{|I|}}$ of the nonempty trees $\mathcal{Q}_{i_1}, \mathcal{Q}_{i_2}, \dots, \mathcal{Q}_{i_{|I|}}$, respectively, and so $ch_{\mathcal{Q}}(b) = |I| \leq ch(b)$. In addition, b has no parent in \mathcal{Q} , and so it is the root of \mathcal{Q} . \square

For a tree τ , the root $rt(\tau)$ of τ and its left-most vertex $l(\tau)$ are called the *sentinels* of τ . The next lemma shows that each subtree in the forest $T \setminus C$ is incident to at most two cut vertices. The proof of this lemma follows lines similar to those of the proof of Lemma 2.14, and is thus omitted.

LEMMA 2.16. *For any subtree T' in the forest $T \setminus C$, no other vertex in T' other than its two sentinels $rt(T')$ and $l(T')$ may be incident to a vertex from C . Moreover, both $rt(T')$ and $l(T')$ are incident to at most one vertex from C ; specifically, $rt(T')$ is incident to its parent in T unless $rt(T')$ is the root of T , and $l(T')$ is incident to its left-most child in T unless $l(T')$ is a leaf in T .*

Similarly to the 1-dimensional case, we add the two sentinels $rt(T)$ and $l(T)$ of the original tree T to the set C of cut vertices. From now on we refer to the appended set $\tilde{C} = C \cup \{rt(T), l(T)\}$ as the set of *cut vertices*. Intuitively, Lemma 2.16 shows that the Procedure *CV* “slices” the tree in a “path-like” fashion, i.e., in a way that is analogous to the decomposition of ϑ_n into intervals described in section 2.2.1. (See Figure 6 for an illustration.)

Lemmas 2.12, 2.14, 2.15, and 2.16 imply the following corollary, which summarizes the properties of the set \tilde{C} of cut vertices.

COROLLARY 2.17.

1. For $n \geq 2d$, $|\tilde{C}| \leq (n/d) + 1$.
2. The size of any subtree in the forest $T \setminus \tilde{C}$ is smaller than $2d$.
3. $\tilde{\mathcal{Q}} = \mathcal{Q}(T, \tilde{C})$ is a spanning tree of \tilde{C} rooted at $rt(T)$, with $\Delta(\tilde{\mathcal{Q}}) \leq \Delta(T)$.
4. For any subtree T' in the forest $T \setminus \tilde{C}$, only the two sentinels $rt(T')$ and $l(T')$ of T' are incident to a vertex from \tilde{C} . Moreover, both $rt(T')$ and $l(T')$ are incident to at most one vertex from \tilde{C} ; specifically, $rt(T')$ is incident to its parent in T , and $l(T')$ is incident to its left-most child in T unless $l(T')$ is a leaf in T .

Remark The running time of the procedure *CV* is $O(n)$. Hence the set \tilde{C} of cut vertices can be computed in linear time.

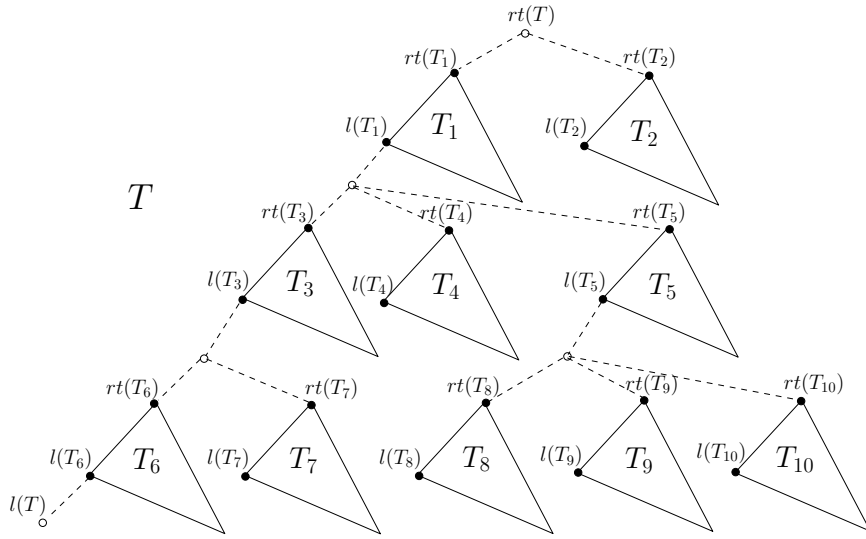


FIG. 6. A path-like decomposition of the tree T into subtrees T_1, T_2, \dots, T_{10} . The five cut vertices of \tilde{C} (i.e., the three vertices of C and the two sentinels $rt(T)$ and $l(T)$ of T) are depicted in the figure by empty dots, whereas the 20 sentinels of the subtrees T_1, T_2, \dots, T_{10} are depicted by filled dots. Similarly to the 1-dimensional case, each subtree T_i is incident to at most two cut vertices. Edges in T that connect sentinels of subtrees with cut vertices are depicted by dashed lines.

2.3.2. 1-spanners with low diameter. Consider an n -vertex (weighted) tree T , and let M_T be the tree metric induced by T . In this section we devise a construction $\mathcal{H}_k(n)$ of 1-spanners for M_T with comparable monotone diameter $\bar{\Lambda}(n) = \bar{\Lambda}(\mathcal{H}_k(n))$ in the range $\Omega(\alpha(n)) = \bar{\Lambda}(n) = O(\log n)$. Both in this construction and in the one of section 2.3.3, all edges of the original tree T are added to the spanner.

Let k be a fixed parameter such that $4 \leq k \leq n/2 - 1$, and set $d = n/k$. (Observe that $n \geq 2k + 2$ and $d > 2$.) To select the set \tilde{C} of cut vertices, we invoke the procedure CV on the input (T, rt) and d . Set $C = CV((T, rt), d)$ and $\tilde{C} = C \cup \{rt(T), l(T)\}$. Since $k \geq 4$, it holds that $2d = 2n/k < n$. Denote the subtrees in the forest $T \setminus \tilde{C}$ by T_1, T_2, \dots, T_p . By Corollary 2.17, $|\tilde{C}| \leq (n/d) + 1 = k + 1$, and each subtree T_i in $T \setminus \tilde{C}$ has size less than $2d = 2n/k$. Observe that $\sum_{i=1}^p |T_i| = n - |\tilde{C}| \geq n - k - 1$, implying that the number p of subtrees in $T \setminus \tilde{C}$ satisfies

$$(2.3) \quad p \geq \frac{n - k - 1}{2n/k} \geq \frac{k}{4}.$$

(The last inequality holds for $k \leq n/2 - 1$.)

To connect the set \tilde{C} of cut vertices, the algorithm first constructs the tree $\tilde{Q} = Q(T, \tilde{C})$. Observe that \tilde{Q} inherits the tree structure of T ; that is, for any two points u and v in \tilde{C} , u is an ancestor of v in \tilde{Q} if and only if it is its ancestor in T . Consequently, any 1-spanner path for the tree metric $M_{\tilde{Q}}$ induced by \tilde{Q} between two arbitrary comparable⁷ points is also a 1-spanner path for the original tree metric M_T . The algorithm proceeds by building a 1-spanner for \tilde{Q} via one of the aforementioned generalized constructions from [14, 2, 49, 41] (henceforth, *tree-spanner*). (See the first paragraph of section 1.2 for more details.) In other words, $O(k)$ edges between cut

⁷This may not hold true for two points that are not comparable, as their least common ancestor may not belong to \tilde{Q} .

vertices are added to the spanner $\mathcal{H}_k(n)$ to guarantee that the monotone distance in the spanner between any two comparable cut vertices will be $O(\alpha(k))$. Then the algorithm adds to the spanner $\mathcal{H}_k(n)$ edges that connect each of the two sentinels to all other cut vertices. (In fact, the leaf $l(T)$ need not be connected to all cut vertices but rather only to those which are its ancestors in T .) Finally, the algorithm calls itself recursively for each of the subtrees T_1, T_2, \dots, T_p of T . At the bottom level of the recursion, i.e., when $n < 2k + 2$, the algorithm uses the tree-spanner to connect all points, and, in addition, it adds to the spanner edges that connect each of the two sentinels $rt(T)$ and $l(T)$ to all the other $n - 1$ points.

We next analyze the properties of the constructed spanner $\mathcal{H}_k(n)$.

LEMMA 2.18. *The number of edges in $\mathcal{H}_k(n)$ is at most $O(n)$.*

Proof. We denote by $E(n)$ the number of edges in $\mathcal{H}_k(n)$, excluding edges of T . Clearly, $E(n)$ satisfies the recurrence $E(n) \leq O(k) + \sum_{i=1}^p E(|T_i|)$, with the base condition $E(q) = O(q)$ for all $q < 2k + 2$. Recall that for each $i \in [p]$, $|T_i| \leq 2d = 2n/k < n$, and by (2.3) we have $p \geq k/4$. Also, since \tilde{C} is nonempty, it holds that $\sum_{i=1}^p |T_i| = n - |\tilde{C}| \leq n - 1$. Next, we prove by induction on n that $E(n) \leq 4c(n - 1)$ for a sufficiently large constant c . The basis $n < 2k + 2$ is immediate. For $n \geq 2k + 2$, the induction hypothesis implies that

$$\begin{aligned} E(n) &\leq c \cdot k + 4c \cdot \sum_{i=1}^p (|T_i| - 1) = c \cdot k - 4c \cdot p + 4c \cdot \sum_{i=1}^p |T_i| \\ &\leq c(k - 4p) + 4c(n - 1) \leq 4c(n - 1). \end{aligned}$$

(The last inequality holds as $p \geq k/4$.) Including edges of T , the number of edges increases by at most $n - 1$ units. \square

LEMMA 2.19. *The maximum degree $\Delta(\mathcal{H}_k(n))$ of $\mathcal{H}_k(n)$ is at most $2k + \Delta(T)$.*

Proof. Denote by $\Delta(n)$ the maximum degree of a vertex in $\mathcal{H}_k(n)$, excluding edges of T . Since $|\tilde{C}| \leq k + 1$, $\Delta(n)$ satisfies the recurrence $\Delta(n) \leq \max\{k, \Delta(2n/k)\}$, with the base condition $\Delta(q) \leq 2k$ for all $q < 2k + 2$. It follows that $\Delta(n) \leq 2k$. Including edges of T , the maximum degree increases by at most $\Delta(T)$ units. \square

LEMMA 2.20. *The lightness $\Psi(\mathcal{H}_k(n))$ of $\mathcal{H}_k(n)$ is at most $O(k \cdot \log_k n)$.*

Proof. To prove the lemma we extend the notion of *load* defined in [1] for 1-dimensional spaces to general tree metrics.⁸ Consider an edge $e' = (v, w)$ connecting two arbitrary points in M_T , and an edge $e \in E(T)$. The edge e' is said to *load* e if the unique path in T between the endpoints v and w of e' traverses e . For a spanning subgraph H of M_T the number of edges $e' \in E(H)$ that load an edge $e \in E(T)$ is called the *load* of e by H and is denoted by $\chi(e) = \chi_H(e)$. The *load* of H (with respect to T), $\chi(H) = \chi_T(H)$, is the maximum load of an edge of T by H . Recall that H is a 1-spanner for M_T ; hence for each edge $e' \in E(H)$ we have $w(e') = \sum_{\{e \in E(T) : e \text{ loaded by } e'\}} w(e)$. By double-counting, we get

$$\begin{aligned} w(H) &= \sum_{e' \in E(H)} w(e') = \sum_{e' \in E(H)} \sum_{\{e \in E(T) : e \text{ loaded by } e'\}} w(e) \\ &= \sum_{e \in E(T)} \sum_{\{e' \in E(H) : e' \text{ loads } e\}} w(e) \\ (2.4) \quad &= \sum_{e \in E(T)} \chi_H(e) \cdot w(e) \leq \chi(H) \cdot \sum_{e \in E(T)} w(e) = \chi(H) \cdot w(T), \end{aligned}$$

⁸Agarwal, Wang, and Yin [1] used a slightly different notion, which they called *covering*. The notion of load as defined above was introduced in [22], but the two notions are very close.

implying that $\Psi(H) = w(H)/w(T) \leq \chi(H)$. Thus it suffices to provide an upper bound of $O(k \cdot \log_k n)$ on the load $\chi(\mathcal{H}_k(n))$ of $\mathcal{H}_k(n)$. Denote by $\chi(n)$ the load of $\mathcal{H}_k(n)$, excluding edges of T . After the first level of recursion, $\mathcal{H}_k(n)$ contains only $O(k)$ edges that connect cut vertices. These edges contribute $O(k)$ units of load to each edge of T . In particular, after the first level of recursion, each subtree in the forest $T \setminus \tilde{C}$ is loaded by at most $O(k)$ edges. Hence $\chi(n)$ satisfies the recurrence $\chi(n) \leq O(k) + \chi(2n/k)$, with the base condition $\chi(q) = O(q)$ for all $q < 2k+2$, yielding $\chi(n) = O(k \cdot \log_k n)$. Including edges of T , the load increases by one unit. \square

LEMMA 2.21. *The comparable monotone diameter $\bar{\Lambda}(n) = \bar{\Lambda}(\mathcal{H}_k(n))$ of $\mathcal{H}_k(n)$ is at most $O(\log_k n + \alpha(k))$.*

Proof. The leaf radius $\hat{R}(n)$ of $\mathcal{H}_k(n)$ is defined as the maximum monotone distance between the left-most vertex $l(T)$ in T and one of its ancestors in T . By Corollary 2.17, similarly to the 1-dimensional case, $\hat{R}(n)$ satisfies the recurrence $\hat{R}(n) \leq 2 + \hat{R}(2n/k)$, with the base condition $\hat{R}(q) = 1$ for all $q < 2k+2$. Hence, $\hat{R}(n) = O(\log_k n)$. Similarly, we define the root radius $\check{R}(n)$ as the maximum monotone distance between the root $rt(T)$ of T and some other point in T . By the same argument we get $\check{R}(n) = O(\log_k n)$. Applying again Corollary 2.17 and reasoning similar to the 1-dimensional case, we get that $\bar{\Lambda}(n)$ satisfies the recurrence $\bar{\Lambda}(n) \leq \max\{\bar{\Lambda}(2n/k), O(\alpha(k)) + \hat{R}(2n/k) + \check{R}(2n/k)\}$, with the base condition $\bar{\Lambda}(q) = O(\alpha(q))$ for all $q < 2k+2$. It follows that $\bar{\Lambda}(n) = O(\log_k n + \alpha(k))$. \square

LEMMA 2.22. *The worst-case running time of the algorithm is at most $O(n \cdot \log_k n)$.*

Proof. Denote the worst-case running time of the algorithm by $t(n)$, excluding the time needed to add the edges of T to the spanner. The algorithm starts by invoking the decomposition procedure for selecting the set \tilde{C} of cut vertices. As was mentioned above, this step requires $O(n)$ time. Next, the algorithm builds the tree \tilde{Q} , which can be carried out in time $O(|\tilde{Q}|) = O(k)$. The algorithm proceeds by building the tree-spanner for \tilde{Q} . The tree-spanner of [14, 2, 49, 41] can be built within linear time. Hence, building the tree-spanner for \tilde{Q} requires $O(k)$ time. Next, the algorithm adds to the spanner edges that connect each of the two sentinels to all other cut vertices, which can be carried out within time $O(k)$ as well. Finally, the algorithm calls itself recursively for each of the subtrees T_1, T_2, \dots, T_p of T , which requires at most $\sum_{i=1}^p t(|T_i|)$ time. At the bottom level of the recursion, i.e., when $n < 2k+2$, the algorithm uses the tree-spanner to connect all points, and in addition, it adds to the spanner edges that connect each of the two sentinels of the tree to all the other $n-1$ points. Hence, the running time of the algorithm at the bottom level of the recursion is $O(n)$. It follows that $t(n)$ satisfies the recurrence $t(n) \leq O(n) + \sum_{i=1}^p t(|T_i|)$, with the base condition $t(q) = O(q)$ for all $q < 2k+2$. Recall that $k \geq 4$, and for each $i \in [p]$, $|T_i| \leq 2d = 2n/k < n$. We conclude that $t(n) = O(n \cdot \log_k n)$. Adding the edges of T to the spanner takes another $O(n)$ of time. \square

Summarizing, we have proved the following theorem.

THEOREM 2.23. *For any tree metric M_T and a parameter $k \geq 4$ there exists a 1-spanner with maximum degree at most $\Delta(T) + 2k$, diameter $O(\log_k n + \alpha(k))$, lightness $O(k \cdot \log_k n)$, and $O(n)$ edges. The running time of this construction is $O(n \cdot \log_k n)$.*

Remark. The maximum degree $\Delta(\mathcal{H})$ of the spanner $\mathcal{H} = \mathcal{H}_k(n)$ cannot be in general smaller than the maximum degree $\Delta(T)$ of the original tree. Indeed, consider a unit weight star T with edge set $\{(rt, v_1), (rt, v_2), \dots, (rt, v_{n-1})\}$. Obviously, any spanner \mathcal{H} for M_T with $\Delta(\mathcal{H}) < n-1$ distorts the distance between the root rt and some other vertex.

2.3.3. 1-spanners with high diameter. In this section we devise a construction $\mathcal{H}'_k(n)$ of 1-spanners for M_T with comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}(\mathcal{H}'_k(n))$ in the range $\bar{\Lambda}'(n) = \Omega(\log n)$.

The algorithm starts by constructing the tree $\tilde{Q} = Q(T, \tilde{C})$ that spans the set \tilde{C} of cut vertices. All edges of \tilde{Q} are inserted into $\mathcal{H}'_k(n)$. (This step is analogous to taking the edges $(r_0, r_1), (r_1, r_2), \dots, (r_{k-1}, r_k)$ in the 1-dimensional construction of section 2.2.3.) Observe that the depth of \tilde{Q} is at most k , implying that any two comparable cut vertices are connected via a 1-spanner path in \tilde{Q} that consists of at most k edges; since \tilde{Q} inherits the tree structure of T , this path is also a 1-spanner path for the original tree metric M_T . Then the algorithm calls itself recursively for each of the subtrees T_1, T_2, \dots, T_p of T . At the bottom level of the recursion, i.e., when $n < 2k + 2$, the algorithm adds no additional edges to the spanner.

We next analyze the properties of the constructed spanner $\mathcal{H}'_k(n)$.

The proof of the next lemma is similar to the proof of Lemma 2.18 from section 2.3.2.

LEMMA 2.24. *The number of edges in $\mathcal{H}'_k(n)$ is at most $O(n)$.*

LEMMA 2.25. *The maximum degree $\Delta(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ is at most $2 \cdot \Delta(T)$.*

Proof. Denote by $\Delta'(n)$ the maximum degree of a vertex in $\mathcal{H}'_k(n)$, excluding edges of T , and let $\Delta_0 = \Delta(T)$ denote the maximum degree of the original tree T . By the third assertion of Corollary 2.17, $\Delta(\tilde{Q}) \leq \Delta_0$, and so $\Delta'(n)$ satisfies the recurrence $\Delta'(n) \leq \max\{\Delta_0, \Delta'(2n/k)\}$, with the base condition $\Delta'(q) = 0$ for all $q < 2k + 2$, yielding $\Delta'(n) \leq \Delta_0$. Including edges of T , the maximum degree increases by at most $\Delta(T)$ units. It follows that the maximum degree $\Delta(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ is at most $\Delta_0 + \Delta(T) = 2 \cdot \Delta(T)$. \square

LEMMA 2.26. *The lightness $\Psi(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ is at most $O(\log_k n)$.*

Proof. By (2.4), showing that the load $\chi(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ is bounded by $O(\log_k n)$ will prove the lemma. Denote by $\chi'(n)$ the load of $\mathcal{H}'_k(n)$, excluding edges of T . After the first level of recursion, $\mathcal{H}'_k(n)$ contains just the edges of the tree \tilde{Q} . We argue that each edge $e = (u, v)$ of T is loaded by at most one edge of \tilde{Q} . Indeed, if both u and v are cut vertices, then e is also an edge of \tilde{Q} , and so it is loaded by itself. Otherwise, either u or v (or both of them) belongs to some subtree T_i in the forest $T \setminus \tilde{C}$. In this case, the fourth assertion of Corollary 2.17 implies that e is loaded by at most one edge in \tilde{Q} , namely, the edge connecting the parent of $rt(T_i)$ in T and the left-most child of $l(T_i)$ in T , if it exists. In particular, after the first level of recursion, each subtree in the forest $T \setminus \tilde{C}$ is loaded by at most one edge. Hence $\chi'(n)$ satisfies the recurrence $\chi'(n) \leq 1 + \chi'(2n/k)$, with the base condition $\chi'(q) = 0$ for all $q < 2k + 2$. It follows that $\chi'(n) = O(\log_k n)$. Including edges of T , the load increases by one unit. \square

LEMMA 2.27. *The comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ is at most $O(\log_k n + \alpha(k))$.*

Proof. By Corollary 2.17, similarly to the 1-dimensional case, the leaf radius $\hat{R}'(n)$ of $\mathcal{H}'_k(n)$ satisfies the recurrence $\hat{R}'(n) \leq k + \hat{R}'(2n/k)$, with the base condition $\hat{R}'(q) \leq q - 1$ for all $q < 2k + 2$, yielding $\hat{R}'(n) = O(k \cdot \log_k n)$. Similarly, we get that $\check{R}'(n) = O(k \cdot \log_k n)$. Applying Corollary 2.17 and reasoning similar to the 1-dimensional case, we get that the comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ satisfies the recurrence $\bar{\Lambda}'(n) \leq \max\{\bar{\Lambda}'(2n/k), k + \check{R}'(2n/k) + \hat{R}'(2n/k)\}$, with the base condition $\bar{\Lambda}'(q) \leq q - 1$ for all $q < 2k + 2$. It follows that $\bar{\Lambda}'(n) = O(k \cdot \log_k n)$. \square

The proof of the next lemma is similar to the proof of Lemma 2.22 from section 2.3.2.

LEMMA 2.28. *The worst-case running time of the algorithm is at most $O(n \cdot \log_k n)$.*

Finally, we remark that $\mathcal{H}'_k(n)$ is a planar graph. Summarizing, we have proved the following theorem.

THEOREM 2.29. *For any tree metric M_T and a parameter k there exists a 1-spanner with maximum degree at most $2 \cdot \Delta(T)$, diameter $O(k \cdot \log_k n)$, lightness $O(\log_k n)$, and $O(n)$ edges. Moreover, this 1-spanner is a planar graph. The running time of this construction is $O(n \cdot \log_k n)$.*

3. Euclidean spanners. In this section we demonstrate that our 1-spanners for tree metrics can be used for constructing Euclidean spanners. More specifically, we employ the dumbbell theorem of [4] in conjunction with our 1-spanners for tree metrics to construct Euclidean spanners.

THEOREM 3.1 (dumbbell theorem, Theorem 2 in [4]). *Given a set S of n points in \mathbb{R}^d and a parameter $\epsilon > 0$, a forest \mathcal{D} consisting of $O(1)$ rooted binary trees each of size $O(n)$ can be built in time $O(n \cdot \log n)$, having the following properties:*

1. *For each tree in \mathcal{D} there is a 1-1 correspondence between the leaves of this tree and the points of S .*
2. *Each internal vertex in the tree has a unique representative point, which can be selected arbitrarily from the points in any of its descendant leaves.*
3. *Consider a tree T and two leaves u' and v' in T . Let P' be the unique path in T connecting u' and v' , and let P be the path obtained from P' by replacing every vertex $y' \in P'$ by its representative y .*

The forest \mathcal{D} is required to satisfy the following condition: for every two points $u, v \in S$ there is a tree T in \mathcal{D} such that the path P obtained in the above way from the unique path in T between the two leaves u' and v' corresponding to u and v is a $(1 + \epsilon)$ -spanner path for u and v .

For each dumbbell tree in \mathcal{D} , we use the following representative assignment from [4]. Leaf labels are propagated up the tree. An internal vertex chooses for itself one of the propagated labels and propagates the other one up the tree. Each label is used at most twice, once at a leaf and once at an internal vertex. Any label assignment induces a weight function over the edges of the dumbbell tree in the obvious way. Specifically, the weight of an edge is set to be the Euclidean distance between the representatives corresponding to the two endpoints of that edge. Arya et al. [4] proved that the lightness of dumbbell trees is always $O(\log n)$, regardless of which representative assignment is chosen for the internal vertices.

Next, we describe our construction of Euclidean spanners with diameter in the range $\Omega(\alpha(n)) = \Lambda = O(\log n)$.

We note that each dumbbell tree has size $O(n)$. For each (weighted) dumbbell tree $DT_i \in \mathcal{D}$, denote by M_i the $O(n)$ -point tree metric induced by DT_i . To obtain our construction of $(1 + \epsilon)$ -spanners with low diameter, we set $k = n^{1/\Lambda}$ and build the 1-spanner construction $\mathcal{H}^i = \mathcal{H}_k^i(O(n))$ that is guaranteed by Theorem 2.23 for each of the tree metrics M_i . (Observe that the parameter k controls the maximum degree of the resulting spanner. To obtain diameter Λ it is natural to set $k = n^{1/\Lambda}$.) Then we translate each \mathcal{H}^i to be a spanning subgraph $\check{\mathcal{H}}^i$ of S in the following way: each edge in \mathcal{H}^i is replaced with an edge that connects the representatives corresponding to the endpoints of that edge. Finally, let $\mathcal{E}_k(n)$ be the spanner obtained from the union of all the graphs $\check{\mathcal{H}}^i$.

Theorem 2.23 implies that each graph $\check{\mathcal{H}}^i$ contains only $O(n)$ edges. By the dumbbell theorem, $\mathcal{E}_k(n)$ is the union of a constant number of such graphs. Thus the

total number of edges in $\mathcal{E}_k(n)$ is $O(n)$.

We proceed by showing that $\mathcal{E}_k(n)$ is a $(1 + \epsilon)$ -spanner for S with diameter $\Lambda = \Lambda(\mathcal{E}_k(n))$ at most $O(\log_k n + \alpha(k))$. By the dumbbell theorem, for any pair u, v of points in S there is a dumbbell tree DT_i , so that the unique path $P_{u,v}$ connecting u and v in DT_i is a $(1 + \epsilon)$ -spanner path for them. Theorem 2.23 implies that there is a 1-spanner path P in \mathcal{H}^i between u and v that consists of at most $O(\log_k n + \alpha(k))$ edges. By the triangle inequality, the weight of the corresponding translated path \check{P} in $\check{\mathcal{H}}^i$ is no greater than the weight of $P_{u,v}$. Hence, \check{P} is a $(1 + \epsilon)$ -spanner path for u and v that consists of at most $O(\log_k n + \alpha(k))$ edges.

Next, we show that the maximum degree $\Delta(\mathcal{E}_k(n))$ of $\mathcal{E}_k(n)$ is $O(k)$. Since each dumbbell tree DT_i is binary, Theorem 2.23 implies that $\Delta(\mathcal{H}^i) = O(k)$. Recall that each label is used at most twice in DT_i , and so $\Delta(\check{\mathcal{H}}^i) \leq 2 \cdot \Delta(\mathcal{H}^i) = O(k)$. The union of $O(1)$ such graphs will also have maximum degree $O(k)$.

We argue that the lightness $\Psi(\mathcal{E}_k(n))$ of $\mathcal{E}_k(n)$ is $O(k \cdot \log_k n \cdot \log n)$. Consider an arbitrary dumbbell tree DT_i . Recall that the lightness of all dumbbell trees is $O(\log n)$, and so $w(DT_i) = O(\log n) \cdot w(MST(S))$. By Theorem 2.23, the weight $w(\mathcal{H}^i)$ of \mathcal{H}^i is at most $O(k \cdot \log_k n) \cdot w(DT_i) = O(k \cdot \log_k n \cdot \log n) \cdot w(MST(S))$. By the triangle inequality, the weight of each edge in $\check{\mathcal{H}}^i$ is no greater than the corresponding weight in \mathcal{H}^i , implying that the weight $w(\check{\mathcal{H}}^i)$ of the graph $\check{\mathcal{H}}^i$ satisfies $w(\check{\mathcal{H}}^i) \leq w(\mathcal{H}^i) = O(k \cdot \log_k n \cdot \log n) \cdot w(MST(S))$. The union of $O(1)$ such graphs will also have weight $O(k \cdot \log_k n \cdot \log n) \cdot w(MST(S))$.

Finally, we bound the running time of this construction. By the dumbbell theorem, the forest \mathcal{D} of dumbbell trees can be built in $O(n \cdot \log n)$ time. Theorem 2.23 implies that we can compute each of the graphs \mathcal{H}^i in time $O(n \cdot \log_k n) = O(n \cdot \log n)$. Moreover, as each graph \mathcal{H}^i contains only $O(n)$ edges, translating it into a graph $\check{\mathcal{H}}^i$ as described above can be carried out in $O(n)$ time. Since there is a constant number of such graphs, it follows that the overall time needed to compute our construction $\mathcal{E}_k(n)$ of Euclidean spanners is $O(n \cdot \log n)$.

To obtain our construction of Euclidean spanners for the complementary range $\Lambda = \Omega(\log n)$, we use our 1-spanners for tree metrics from Theorem 2.29 instead of Theorem 2.23.

COROLLARY 3.2. *For any set S of n points in \mathbb{R}^d , any $\epsilon > 0$, and a parameter k , there exists a $(1 + \epsilon)$ -spanner with maximum degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, lightness $O(k \cdot \log_k n \cdot \log n)$, and $O(n)$ edges. There also exists a $(1 + \epsilon)$ -spanner with maximum degree $O(1)$, diameter $O(k \cdot \log_k n)$, and lightness $O(\log_k n \cdot \log n)$. Both these constructions can be implemented in time $O(n \cdot \log n)$.*

4. Spanners for doubling metrics. In this section we demonstrate that our 1-spanners for tree metrics can be used to construct spanners for doubling metrics. Our approach in this section is similar to that used in section 3 (the Euclidean case), but instead of using dumbbell trees we use the net-tree skeletons of the construction by Gottlieb and Roditty [26].

Let $M = (P, \delta)$ be an n -point doubling metric. A $(1 + \epsilon)$ -spanner H for M is called a *tree-like spanner* if it contains a tree T that satisfies the following conditions:

1. Each vertex v of T is assigned a representative point $r(v) \in P$.
2. There is a 1-1 correspondence between the points of P and the representatives of the leaves of T .
3. Each internal vertex is assigned a unique representative. (Thus, each point of P will be the representative of at most two vertices of T .) In particular, there are at most $2n$ vertices in T .

4. For any two points $p, q \in P$ there is a $(1 + \epsilon)$ -spanner path in H between p and q that is composed of three consecutive parts: (a) a path ascending the edges of T , (b) a single edge, and (c) a path descending the edges of T . (Each edge $e = (u, v)$ in T is translated into an edge $(r(u), r(v))$ in H .)

We say that such a tree T is a *tree-skeleton* of the spanner H .

Gottlieb and Roditty [26] proved the following theorem.

THEOREM 4.1 (see [26]). *For any n -point doubling metric $M = (P, \delta)$ and any $\epsilon > 0$ one can build in $O(n \cdot \log n)$ time a $(1 + \epsilon)$ -spanner H and a tree-skeleton T for H such that both H and T have constant degree.*

The drawback of the spanner of Gottlieb and Roditty [26] is that its diameter may be arbitrarily large. However, we can employ Theorem 2.23 to reduce the diameter of this spanner.

Next, we describe a spanner construction H^* for doubling metrics with small degree and diameter in the range $\Omega(\alpha(n)) = \Lambda = O(\log n)$.

We start by building the spanner H and its tree-skeleton T that are guaranteed by Theorem 4.1. Note that T contains at most $2n = O(n)$ vertices. Next, we set $k = n^{1/\Lambda}$ and build the 1-spanner G_k for the tree metric $M_T = (P, \delta_T)$ induced by T that is guaranteed by Theorem 2.23. Notice that the edge weights of G_k are assigned according to the distance function δ_T of the tree metric M_T . The 1-spanner G_k is converted into a graph G_k^* over the point set P in the following way: each edge (u, v) of G_k , for a pair u, v of vertices in T , is translated into the edge $(r(u), r(v))$ between their corresponding representatives. Finally, let H^* be the spanner obtained from the union of the graphs H and G_k^* .

Next, we analyze the properties of the spanner H^* .

By Theorems 2.23 and 4.1, the graphs H and G_k^* contain $O(n)$ edges each, and they can be built in $O(n \cdot \log n)$ time. Hence the same bounds $O(n)$ and $O(n \cdot \log n)$ on the number of edges and running time, respectively, of these constructions apply to their union H^* .

Next, we bound the maximum degree $\Delta(H^*)$ of H^* . Clearly $\Delta(H^*) \leq \Delta(H) + \Delta(G_k^*)$. By Theorem 4.1, we have $\Delta(T) = O(1)$, $\Delta(H) = O(1)$. Also, Theorem 2.23 yields $\Delta(G_k) \leq \Delta(T) + 2k = O(k)$. Since each point of P is assigned as the representative of at most two vertices of T , it follows that $\Delta(G_k^*) \leq 2 \cdot \Delta(G_k) = O(k)$. Altogether, we have $\Delta(H^*) \leq \Delta(H) + \Delta(G_k^*) = O(k)$.

Finally, we show that H^* is a $(1 + \epsilon)$ -spanner for M with diameter $\Lambda(H) = O(\log_k n + \alpha(k))$. Consider an arbitrary pair $p, q \in P$ of points, and let u (resp., v) be the leaf vertex of T whose representative is p (resp., q). Since T is a tree-skeleton of H , there is a $(1 + \epsilon)$ -spanner path $\Pi(p, q)$ in H between p and q that is composed of three consecutive parts: (a) a path $\Pi(p, p') = (r(u) = p, \dots, r(u') = p')$, ascending the edges of T ; (b) a single edge (p', q') ; and (c) a path $\Pi(q', q) = (r(v') = q', \dots, r(v) = q)$, descending the edges of T ; the vertex u' (resp., v') is an ancestor of u (resp., v) in T , and $p' = r(u')$ (resp., $q' = r(v')$) is its representative. We have $\Pi(p, q) = \Pi(p, p') \circ (p', q') \circ \Pi(q', q)$, and so

$$(4.1) \quad \omega(\Pi(p, q)) = \omega(\Pi(p, p')) + \delta(p', q') + \omega(\Pi(q', q)) \leq (1 + \epsilon) \cdot \delta(p, q).$$

Theorem 2.23 implies that there is a 1-spanner path $P(u, u')$ (resp., $P(v', v)$) in G_k between u and u' (resp., v' and v) that consists of at most $O(\log_k n + \alpha(k))$ edges. The corresponding translated path $P^*(p, p')$ (resp., $P^*(q', q)$) is a path in G_k^* (and thus in H^*) between the corresponding representatives p and p' (resp., q' and q), having the same number of edges as the 1-spanner path $P(u, u')$ (resp., $P(v', v)$) in G_k ,

namely, $O(\log_k n + \alpha(k))$. Moreover, by the triangle inequality, the weight of the translated path $P^*(p, p')$ (resp., $P^*(q', q)$) is no greater than the weight of the 1-spanner path $P(u, u')$ (resp., $P(v', v)$), which is, in turn, equal to the weight of the original path $\Pi(p, p')$ (resp., $\Pi(q', q)$) in H . Hence $\omega(P^*(p, p')) \leq \omega(P(u, u')) = \omega(\Pi(p, p'))$, $\omega(P^*(q', q)) \leq \omega(P(v', v)) = \omega(\Pi(q', q))$. Let $P^*(p, q)$ be the path obtained as the concatenation of the path $P^*(p, p')$, the edge (p', q') , and the path $P^*(q', q)$; i.e., $P^*(p, q) = P^*(p, p') \circ (p', q') \circ P^*(q', q)$. It is easy to see that $P^*(p, q)$ is a path in H^* between p and q , having $|P^*(p, q)| = |P^*(p, p')| + 1 + |P^*(q', q)| = O(\log_k n + \alpha(k))$ edges. Also, the weight $\omega(P^*(p, q))$ of $P^*(p, q)$ satisfies

$$\begin{aligned} \omega(P^*(p, q)) &= \omega(P^*(p, p')) + \delta(p', q') + \omega(P^*(q', q)) \\ &\leq \omega(\Pi(p, p')) + \delta(p', q') + \omega(\Pi(q', q)) \leq (1 + \epsilon) \cdot \delta(p, q). \end{aligned}$$

(The last inequality follows from (4.1).) In other words, $P^*(p, q)$ is a $(1 + \epsilon)$ -spanner path in H^* between p and q that consists of $O(\log_k n + \alpha(k))$ edges.

COROLLARY 4.2. *For any n -point doubling metric $M = (P, \delta)$, any $\epsilon > 0$, and a parameter k there exists a $(1 + \epsilon)$ -spanner with maximum degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, and $O(n)$ edges. The running time of this construction is $O(n \cdot \log n)$.*

It can be shown that the spanners constructed in this way have lightness $O(k \cdot \log_k n \cdot \log n)$. Moreover, as in the Euclidean case, we can also achieve the complementary trade-off between the diameter and lightness for doubling metrics. To this end, we use our 1-spanners for tree metrics from Theorem 2.29 instead of Theorem 2.23 to shortcut the tree metric M_T induced by the tree-skeleton T . Specifically, for a parameter k this trade-off gives rise to degree $O(1)$, diameter $O(k \cdot \log_k n)$, and lightness $O(\log_k n \cdot \log n)$. The proof of these statements is pretty straightforward, and we omit it. In our recent work [23] we devised a transformation that converts $(1 + \epsilon)$ -spanners with maximum degree $O(k)$ and diameter $O(\log_k n + \alpha(k))$ but without any guarantee on their lightness (i.e., the spanners provided by Corollary 4.2) into spanners with the same (up to constants) degree and diameter, but with optimal lightness $O(k \cdot \log_k n)$. Given this result, it appears that analyzing the lightness of the spanners that we constructed in this section is redundant. (The lightness of the spanners constructed in this section is suboptimal by a factor of $\log n$, in contrast to the optimal lightness of the spanners achieved via the transformation of [23].)

Acknowledgments. We are grateful to Sunil Arya, David Mount, and Michiel Smid for helpful discussions.

REFERENCES

- [1] P. K. AGARWAL, Y. WANG, AND P. YIN, *Lower bound for sparse Euclidean spanners*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2005, pp. 670–671.
- [2] N. ALON AND B. SCHIEBER, *Optimal Preprocessing for Answering On-line Product Queries*, manuscript, 1987.
- [3] I. ALTHÖFER, G. DAS, D. P. DOBKIN, D. JOSEPH, AND J. SOARES, *On sparse spanners of weighted graphs*, Discrete Comput. Geom., 9 (1993), pp. 81–100.
- [4] S. ARYA, G. DAS, D. M. MOUNT, J. S. SALOWE, AND M. H. M. SMID, *Euclidean spanners: Short, thin, and lanky*, in Proceedings of the 27th Annual ACM Symposium on Theory of Computing, ACM, New York, 1995, pp. 489–498.
- [5] S. ARYA AND M. H. M. SMID, *Efficient construction of a bounded degree spanner with low weight*, Algorithmica, 17 (1997), pp. 33–54.

- [6] P. BERMAN, A. BHATTACHARYYA, E. GRIGORESCU, S. RASKHODNIKOVA, D. P. WOODRUFF, AND G. YAROSLAVTSEV, *Steiner transitive-closure spanners of low-dimensional posets*, in Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP), Springer, New York, 2011, pp. 760–772.
- [7] A. BHATTACHARYYA, E. GRIGORESCU, M. JHA, K. JUNG, S. RASKHODNIKOVA, AND D. P. WOODRUFF, *Lower bounds for local monotonicity reconstruction from transitive-closure spanners*, in Proceedings of APPROX-RANDOM 2010, Lecture Notes in Comput. Sci., Springer, New York, 2010, pp. 448–461.
- [8] A. BHATTACHARYYA, E. GRIGORESCU, K. JUNG, S. RASKHODNIKOVA, AND D. P. WOODRUFF, *Transitive-closure spanners*, in Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2009, pp. 932–941.
- [9] H. L. BODLAENDER, G. TEL, AND N. SANTORO, *Trade-offs in non-reversing diameter*, Nord. J. Comput., 1 (1994), pp. 111–134.
- [10] T.-H. H. CHAN AND A. GUPTA, *Small hop-diameter sparse spanners for doubling metrics*, in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2006, pp. 70–78.
- [11] H. T.-H. CHAN, A. GUPTA, B. M. MAGGS, AND S. ZHOU, *On hierarchical routing in doubling metrics*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2005, pp. 762–771.
- [12] T.-H. H. CHAN, M. LI, AND L. NING, *Incubators vs. Zombies: Fault-Tolerant, Short, Thin and Lanky Spanners for Doubling Metrics*, Technical Report, CoRR, abs/1207.0892, 2012.
- [13] T.-H. H. CHAN, M. LI, L. NING, AND S. SOLOMON, *New doubling spanners: Better and simpler*, in Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP), Springer, New York, 2013, vol. 1, pp. 315–327.
- [14] B. CHAZELLE, *Computing on a free tree via complexity-preserving mappings*, Algorithmica, 2 (1987), pp. 337–361.
- [15] B. CHAZELLE AND B. ROSENBERG, *The complexity of computing partial sums off-line*, Int. J. Comput. Geom. Appl., 1 (1991), pp. 33–45.
- [16] L. P. CHEW, *There is a planar graph almost as good as the complete graph*, in Proceedings of the 2nd Annual ACM Symposium on Computational Geometry (SOCG), ACM, New York, 1986, pp. 169–177.
- [17] T. H. CORMAN, C. E. LEISERSON, R. L. RIVEST, AND C. STEIN, *Introduction to Algorithms*, 2nd ed., McGraw-Hill, Boston, MA, 2001.
- [18] G. DAS, P. J. HEFFERNAN, AND G. NARASIMHAN, *Optimally sparse spanners in 3-dimensional Euclidean space*, in Proceedings of the 9th Annual ACM Symposium on Computational Geometry (SOCG), ACM, New York, 1993, pp. 53–62.
- [19] G. DAS AND D. JOSEPH, *Which triangulations approximate the complete graph?*, in Proceedings of the International Symposium on Optimal Algorithms, Lecture Notes in Comput. Sci. 401, Springer, New York, 1989, pp. 168–192.
- [20] G. DAS AND G. NARASIMHAN, *A fast algorithm for constructing sparse Euclidean spanners*, in Proceedings of the 10th Annual ACM Symposium on Computational Geometry (SOCG), ACM, New York, 1994, pp. 132–139.
- [21] G. DAS, G. NARASIMHAN, AND J. SALOWE, *A new way to weigh malnourished Euclidean graphs*, in Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1995, pp. 215–222.
- [22] Y. DINITZ, M. ELKIN, AND S. SOLOMON, *Low-light trees, and tight lower bounds for Euclidean spanners*, Discrete Comput. Geom., 43 (2010), pp. 736–783.
- [23] M. ELKIN AND S. SOLOMON, *Optimal Euclidean spanners: Really short, thin and lanky*, in Proceedings of the 45th ACM Symposium on the Theory of Computing, ACM, New York, 2013, pp. 645–654.
- [24] G. N. FREDERICKSON, *A data structure for dynamically maintaining rooted trees*, in Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 1993, pp. 175–184.
- [25] L. GOTTLIEB, A. KONTOROVICH, AND R. KRAUTHGAMER, *Efficient regression in metric space via approximate Lipschitz extension*, in Proceedings of the International Workshop on Similarity-Based Pattern Recognition (SIMBAD), 2013, pp. 43–58.
- [26] L. GOTTLIEB AND L. RODITTY, *An optimal dynamic spanner for doubling metric spaces*, in Proceedings of the 16th European Symposium on Algorithms (ESA), Lecture Notes in Comput. Sci. 5193, Springer, New York, 2008, pp. 478–489.
- [27] J. GUDMUNDSSON, C. LEVCOPOULOS, AND G. NARASIMHAN, *Fast greedy algorithms for constructing sparse geometric spanners*, SIAM J. Comput., 31 (2002), pp. 1479–1500.

- [28] J. GUDMUNDSSON, C. LEVCOPOULOS, G. NARASIMHAN, AND M. SMID, *Approximate distance oracles for geometric graphs*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2002, pp. 828–837.
- [29] J. GUDMUNDSSON, C. LEVCOPOULOS, G. NARASIMHAN, AND M. H. M. SMID, *Approximate distance oracles for geometric spanners*, ACM Trans. Algorithms, 4 (2008), 10.
- [30] J. GUDMUNDSSON, G. NARASIMHAN, AND M. H. M. SMID, *Fast pruning of geometric spanners*, in Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Comput. Sci. 3404, Springer, New York, 2005, pp. 508–520.
- [31] S. HAR-PELED AND M. MENDEL, *Fast construction of nets in low dimensional metrics, and their applications*, in Proceedings of the 21st Annual ACM Symposium on Computational Geometry (SOCG), ACM, New York, 2005, pp. 150–158.
- [32] Y. HASSIN AND D. PELEG, *Sparse communication networks and efficient routing in the plane*, in Proceedings of the 19th ACM Symposium on Principles of Distributed Computing (PODC 2000), ACM, New York, 2000, pp. 41–50.
- [33] J. M. KEIL, *Approximating the complete Euclidean graph*, in Proceedings of the 1st Scandinavian Workshop on Algorithm Theory (SWAT), Springer, New York, 1988, pp. 208–213.
- [34] J. M. KEIL AND C. A. GUTWIN, *Classes of graphs which approximate the complete Euclidean graph*, Discrete Comput. Geom., 7 (1992), pp. 13–28.
- [35] Y. MANSOUR AND D. PELEG, *An approximation algorithm for min-cost network design*, DIMACS Ser. Discrete Math Theoret. Comput. Sci., 53 (2000), pp. 97–106.
- [36] G. NARASIMHAN AND M. SMID, *Geometric Spanner Networks*, Cambridge University Press, Cambridge, UK, 2007.
- [37] D. PELEG AND J. D. ULLMAN, *An optimal synchronizer for the hypercube*, SIAM J. Comput., 18 (1989), pp. 740–747.
- [38] M. PĂTRAȘCU AND E. D. DEMAINE, *Tight bounds for the partial-sums problem*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2004, pp. 20–29.
- [39] S. RAO AND W. D. SMITH, *Approximating geometrical graphs via “spanners” and “banyans,”* in Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1998, pp. 540–550.
- [40] J. RUPPERT AND R. SEIDEL, *Approximating the d-dimensional complete Euclidean graph*, in Proceedings of the 3rd Canadian Conference on Computational Geometry (CCCG’91), 1991, pp. 207–210.
- [41] S. SOLOMON, *An optimal-time construction of sparse Euclidean spanners with tiny diameter*, in Proceedings of the 22st ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2011, pp. 820–839.
- [42] S. SOLOMON, *From hierarchical partitions to hierarchical covers: Optimal fault-tolerant spanners for doubling metrics*, in Proceedings of the 46th Annual ACM Symposium on Theory of Computing, ACM, New York, 2014, pp. 363–372.
- [43] S. SOLOMON, *Fault-Tolerant Spanners for Doubling Metrics: Better and Simpler*, Technical Report, CoRR, abs/1207.7040, 2012.
- [44] S. SOLOMON AND M. ELKIN, *Balancing degree, diameter and weight in Euclidean spanners*, in Proceedings of the 18th European Symposium on Algorithms, 2010, vol. 1, pp. 48–59.
- [45] R. E. TARJAN, *Efficiency of a good but not linear set union algorithm*, J. ACM, 22 (1975), pp. 215–225.
- [46] R. E. TARJAN, *Applications of path compression on balanced trees*, J. ACM, 26 (1979), pp. 690–715.
- [47] M. THORUP, *On shortcutting digraphs*, in Proceedings of the 18th International Workshop on Graph-Theoretic Concepts in Computer Science, 1992, pp. 205–211.
- [48] M. THORUP, *Shortcutting planar digraphs*, Combin. Probab. Comput., 4 (1995), pp. 287–315.
- [49] M. THORUP, *Parallel shortcutting of rooted trees*, J. Algorithms, 23 (1997), pp. 139–159.
- [50] A. C. YAO, *Space-time tradeoff for answering range queries*, in Proceedings of the 14th ACM Symposium on Theory of Computing, ACM, New York, 1982, pp. 128–136.