Balancing Degree, Diameter and Weight in Euclidean Spanners

Shay Solomon and Michael Elkin
Department of Computer Science,
Ben-Gurion University of the Negev
P.O. Box 653, Beer-Sheva 84105, ISRAEL
E-mail: {shayso, elkinm}@cs.bgu.ac.il

Technical Report CS-09-02.

July 23, 2009

Abstract

In this paper we devise a novel *unified* construction that trades between maximum degree, diameter and weight gracefully. For a positive integer parameter k, our construction provides a $(1+\epsilon)$-spanner with maximum degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, weight $O(k \cdot \log_k n \cdot \log n) \cdot w(MST(M))$ and $O(n)$ edges, where α stands for the inverse Ackermann function. Note that for $k = O(1)$ this gives rise to maximum degree $O(1)$, diameter $O(\log n)$ and weight $O(\log^2 n) \cdot w(MST(S))$. 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 $O(n^{1/\alpha(n)})$. 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 $O(n)$. Our bound on the weight is optimal up to a factor of $\log n$.

In addition, in the complementary range of parameters we devise a $(1+\epsilon)$-spanner with maximum degree $O(1)$, diameter $O(k \cdot \log_k n)$ and weight $O(\log_k n \cdot \log n) \cdot w(MST(S))$. This is also optimal up to a factor of $\log n$. Both these constructions can be implemented in time $O(n \cdot \log n)$.

Finally, en route to these results we devise optimal constructions of 1-spanners for general tree metrics, which are of independent interest.

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 x and y. Consider a spanning subgraph $G = (S, E), E \subseteq \binom{S}{2}$ of $\mathcal{S}$, 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 distance $dist_G(x, y)$ between x and y in G is at most $(1 + \epsilon) \cdot \|x - y\|$. Euclidean spanners were introduced[1] more than twenty years ago by Chew [10]. Since then they evolved into an important subarea of Computational Geometry [21, 22, 3, 29, 11, 13, 5, 14, 12, 4, 28, 7, 15]. (See also the recent book on Euclidean spanners [25], and the references therein.) Also, Euclidean spanners have numerous applications in geometric approximation algorithms [28, 24, 17, 18], geometric distance oracles [17, 18, 19], Network Design [20, 23] and in 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*[2] $w(G) = \sum_{e \in E} w(e)$ should not be much greater than $w(MST(S))$. 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] have devised a construction of $(1+\epsilon)$-spanners with lightness $O(\log^2 n)$, diameter $O(\log n)$ and constant degree. They have also devised a construction of $(1 + \epsilon)$-spanners with $O(n)$ (respectively, $O(n \cdot \log^* n)$) edges and diameter $O(\alpha(n))$ (resp., at most 4). However, in the latter construction the resulting spanners may have arbitrarily large lightness and degree. There are also a few other known constructions of $(1 + \epsilon)$-spanners. Das and Narasimhan [13] devised a construction with constant degree and lightness, but the diameter may be arbitrarily large. There is also another construction by Arya et al. [4] that guarantees that both the diameter and the lightness are $O(\log n)$, but 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 this parameter to be arbitrarily large.

In this paper we devise a novel *unified* construction that trades between degree, diameter and weight gracefully. For a positive integer parameter 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 tradeoff 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]. For $k = n^{1/\alpha(n)}$ it gives rise to a spanner with degree $O(n^{1/\alpha(n)})$, diameter $O(\alpha(n))$

[1] The notion "spanner" was coined by Peleg and Ullman [26], who have also introduced spanners for general graphs.

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

	[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	∞	$\log^{\delta} n$	$2^{\sqrt{\log n}}$	$n^{1/\alpha(n)}$	1	1	1
Λ	$\log n$	$\alpha(n)$	$\frac{\log n}{\log\log n}$	$\sqrt{\log n}$	$\alpha(n)$	$\frac{\log^{1+\delta} n}{\log\log n}$	$2^{O(\sqrt{\log n})}$	n^{ζ}
Ψ	$\log^2 n$	∞	$\frac{\log^{2+\delta} n}{\log\log n}$	$2^{O(\sqrt{\log n})}$	$n^{O(1/\alpha(n))}$	$\frac{\log^2 n}{\log\log n}$	$\log^{3/2} n$	$\log n$

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 tradeoffs.) For new results, the second row indicates whether it is obtained by the first (I) or the second (II) tradeoff. (The first tradeoff is degree $O(k)$, diameter $O(\log_k n + \alpha(k))$, and lightness $O(k \cdot \log_k n \cdot \log n)$. The second tradeoff 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 in the corresponding tradeoff. 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. The "∞" symbol means that the respective quantity may be arbitrarily large.

and lightness $O(n^{1/\alpha(n)} \cdot \log n \cdot \alpha(n))$. In the corresponding result from [4] 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 $O(\frac{\log^{1+\delta} n}{\log\log n})$ and lightness $O(\frac{\log^2 n}{\log\log n})$.

All our constructions can be implemented in $O(n \cdot \log n)$ time in the algebraic computation-tree model. (See, e.g., Chapter 3 of [25] for its definition.) This matches the state-of-the-art running time of the aforementioned constructions [4]. See Table 1 for a concise comparison of previous and new results.

Note that in any construction of spanners with degree $O(k)$, the diameter is $\Omega(\log_k n)$. Also, Chan and Gupta [7] 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, than 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 et al. [15] have shown that for any construction of spanners, if the diameter is at most $O(\log_k n)$, then the lightness is at least $\Omega(k \cdot \log_k n)$ and vice versa, if the lightness is at most $O(\log_k n)$, the diameter is at least $\Omega(k \cdot \log_k n)$. This lower bound implies that the bound on lightness in both our tradeoffs cannot possibly be improved by more than a factor of $O(\log n)$. The same slack of $O(\log n)$ is present in the result of [4] that guarantees lightness $O(\log^2 n)$, diameter $O(\log n)$ and constant degree.

1.2 Spanners for Tree Metrics

Let ϑ_n be the metric induced by n points $v_1, v_2, \ldots, v_n$ lying on the x-axis with coordinates $1, 2, \ldots, n$, respectively. In a classical STOC'82 paper [34], Yao showed that there exists a 1-spanner[3] $G = (V, E)$ for ϑ_n with $O(n)$ edges and diameter $O(\alpha(n))$, and that this is tight. Chazelle [8] extended the result of [34] to arbitrary tree metrics. Other proofs of Chazelle's result appeared in [2, 6, 33]. Thorup [33] has also devised an efficient parallel algorithm for computing this 1-spanner. The problem was also studied for planar metrics [32] and for general metrics [31]. (See also Chapter 12 in [25] 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 [30], Yao [34], Chazelle and Rosenberg [9], Pătraşcu and Demaine [27], and the references therein.) For a discussion about the relationship between these two problems see the introduction of [1].

In all constructions [34, 8, 2, 6, 33] 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 lightness follows from the result of [15].

En route to our tradeoffs for Euclidean spanners, we have extended the results of [34, 8, 2, 6, 33] and devised a construction that achieves the *optimal* (up to constant factors) *tradeoff between all involved parameters*. Specifically, consider an n-vertex tree T of degree $\Delta(T)$, and let k be a positive parameter. Our construction provides a 1-spanner for the metric M_T induced by T with $O(n)$ edges, degree $O(k + \Delta(T))$, diameter $O(\log_k n + \alpha(k))$, and lightness $O(k \cdot \log_k n)$. We can also get a spanner with $O(n \cdot \log^* n)$ edges, diameter $O(\log_k n)$, and the same degree and lightness as above. For the complementary range of diameter, another variant of our construction provides a 1-spanner with $O(n)$ edges, degree $O(\Delta(T))$, diameter $O(k \cdot \log_k n)$ and lightness $O(\log_k n)$. As was mentioned above, both tradeoffs are optimal up to constant factors.

We show that this general tradeoff 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. We anticipate that this tradeoff would be found useful in the context of partial sums problems, and for other applications. Finally, we believe that regardless of their applications, our results about spanners for tree metrics are of independent interest.

1.3 Our and Previous Techniques

The starting point for our construction 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 with 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,

[3] The graph G is said to be a *1-spanner* of 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 using the terminology of partial sums. However, the two statements of the problem are equivalent.

to this end Arya et al. [4] employed two main tools. The first one is the *dumbbell trees*, the theory of which was developed by Arya et al. in the same paper [4]. (See also Chapter 11 of [25].) The second one is the bottom-up clustering technique that was developed by Frederickson [16] 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 such that for every pair x, y of points from S there exists a tree $T \in \mathcal{D}$ that satisfies that the distance between x and y in T is a $(1+\epsilon)$-approximation of their Euclidean distance. Arya et al. employ Frederickson's clustering technique on each of these $O(1)$ trees to obtain their ultimate spanner.

Similarly to [4], we start with 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 degree at most 4. (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 with logarithmic diameter and lightness, and with constant degree.) 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 extending 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 tradeoffs in the entire range of parameters. We remark that it is unclear whether the construction of Arya et al. [4] can be extended to provide additional combinations between 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 tradeoffs. Moreover, as can be seen from the discussion above, our construction is much simpler and much 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. We hope that our approach will ultimately help in narrowing the gap of $\log n$ between the upper bound of $O(\log^2 n)$ on the lightness of the spanners in the construction of [4] and the lower bound $\Omega(\log n)$ of [15].

In addition, our construction of 1-spanners for tree metrics is fundamentally different from the previous constructions due to [34, 8, 2, 6, 33]. It is motivated by a recent result of Dinitz et al. [15] that shows that in the context of *spanning trees* one can always trade between the diameter and lightness. The result of [15] suggests that it might be possible to devise 1-spanners for 1-dimensional metrics that exhibit the same optimal tradeoff between the diameter and lightness as the spanning trees of [15] do. While this intuition is generally correct, the construction of [15] is technically quite involved, and it is unclear whether it can be converted into a construction of 1-spanners. To overcome this difficulty, we devise a construction of 1-spanners for 1-dimensional metrics from scratch. Interestingly, our construction also gives rise to a construction of spanning trees that exhibits the same[4] optimal behavior of all involved parameters as the construction of

[4]Actually, the constant factors involved in the tradeoff in the new construction are significantly better than the corresponding factors in [15].

[15]. This new construction is also significantly simpler than that of [15], and we believe that it is of independent interest.

There are standard techniques that enable one to generalize constructions of 1-spanners from 1-dimensional metrics to general tree metrics [8, 2, 6, 33]. These techniques ensure that the upper bound on the diameter of the resulting spanners is 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 [8, 2, 6] to our purposes. Next, we overview this adaptation.

A central ingredient in the generalization techniques of [8, 2, 6] 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 satisfies 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 which is a cut-vertex. For our purposes it is crucial that the degree of the tree $\mathcal{Q}$ will be not (much) greater than the degree of T. In addition, it is essential that each tree $\tau \in \mathcal{F}$ will be incident to at most $O(1)$ cut-vertices. We devise a novel decomposition procedure that guarantees these two 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 diameter, lightness and degree of our constructions for tree metrics (essentially) as small as in the 1-dimensional case.

1.4 Structure of the Paper

In Section 2 we describe our constructions of 1-spanners for tree metrics. Therein we start (Section 2.1) with outlining our basic scheme. We proceed (Section 2.2) with describing our 1-dimensional constructions. In Section 2.3 we extend these constructions to general tree metrics. Our tree decomposition procedure that is in the heart of this extension is described in Section 2.3.1. In Section 3 we derive our results for Euclidean spanners. We leave all issues of running time out of the current version.

1.5 Preliminaries

An *n-point metric space* $M = (V, dist)$ can be viewed as the 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 define the *stretch* between two vertices u and v in V to be $\frac{dist_G(u,v)}{dist(u,v)}$. For any two vertices u, v in a tree T, their (weighted) distance in T is denoted by $dist_T(u, v)$. The tree-metric M_T induced by a tree T is defined as $M_T = (V(T), dist_T)$. For an integer D, a rooted tree in which every vertex has at most D children is called a *D-ary tree*. The *size* of a tree T, denoted $|T|$, is the number of vertices in T.

For a positive integer n, we denote the set $\{1, 2, \ldots, n\}$ 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 an MST of M_T, but its diameter may be huge. We would like to reduce the diameter of this 1-spanner by adding to it as few edges as possible, aiming to preserve both a linear number of edges and small lightness and maximum degree in the process.

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 hops in a 1-spanner path in H connecting 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* (respectively, *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 hops, then any two *arbitrary* points are connected via a 1-spanner path that consists of at most $2h$ hops. Consequently, $\bar{\Lambda}(H) \leq \Lambda(H) \leq 2 \cdot \bar{\Lambda}(H)$. We henceforth restrict the attention to comparable monotone diameter in the sequel.

Let k be a fixed parameter. The first ingredient of the algorithm is to efficiently select a set of $O(k)$ *cut-vertices* whose removal from T decomposes it into a collection of subtrees of size $O(n/k)$ each. (As was mentioned above, we also require this set to satisfy several additional requirements.) Having selected the cut-vertices, the next step of the algorithm is to connect the cut-vertices via $O(k)$ edges, so that any pair of comparable cut-vertices will be reached one from another via a 1-spanner path that consists of a small number of hops. (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 between cut-vertices and subtrees in the spanner. We remark that the spanner contains no other edges that connect between cut vertices and subtrees, or between different subtrees.

2.2 1-Dimensional Spaces

In this section we devise optimal constructions of 1-spanners for ϑ_n. (See Section 1.2 for its definition.) Our argument generalizes easily to any 1-dimensional space.

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

2.2.1 Selecting the Cut Vertices

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, $n \geq k$.) In addition to the two endpoints v_1 and v_n of the path, we select the $k-1$ points $r_1, r_2, \ldots, r_{k-1}$ 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 $r_0 = v_1, r_1, \ldots, r_{k-1}, r_k = v_n$ from the path (along with their incident edges), we are left with k intervals 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 $k = 2$.)

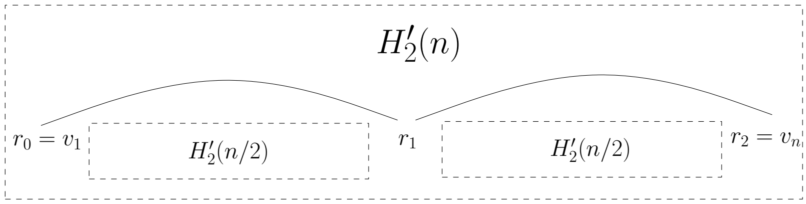

Figure 1: The construction for $k = 2$. Only the first level of the recursion is illustrated. (Edges of the original path P_n are not depicted in the figure.) The cut-vertex $r_1 = v_{n/2}$ is connected via edges to the two sentinels v_1 and v_n. The construction proceeds recursively for each of the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

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 $\Omega(\alpha(n)) = \bar{\Lambda}(n) = O(\log n)$.

First, the algorithm connects the $k+1$ cut-vertices $r_0 = v_1, r_1, \ldots, r_{k-1}, r_k = v_n$ via one of the aforementioned constructions from [34, 8, 2, 6, 33] (henceforth, *list-spanner*). In other words, $O(k)$ edges are added between cut-vertices to guarantee that the monotone distance between any two cut-vertices[5] will be at most $O(\alpha(k))$. Then the algorithm connects each of the two sentinels to all other k cut-vertices. Finally, the algorithm calls itself recursively for each of the intervals. 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 connects both sentinels v_1 and v_n to all the other $n-2$ points. (See Figure 2 for an illustration.)

Denote by $E(n)$ the number of edges in $H_k(n)$, excluding edges of P_n. Clearly, $E(n)$ satisfies the recurrence $E(n) \leq O(k) + k \cdot E(n/k)$, with the base condition $E(n) = O(n)$, for $n \leq k$, yielding $E(n) = O(n)$. 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 $\Delta(n) \leq n-1$, for $n \leq k$, yielding $\Delta(n) \leq k$. Including edges of P_n, the number of edges increases by $n-1$ units, and the maximum degree increases by at most two units.

To bound the weight $w(n) = w(H_k(n))$ of $H_k(n)$, first 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 is at most $w(n/k)$. Observe also that $w(P_n) = n-1$. Hence $w(n)$ satisfies the recurrence $w(n) \leq O(n \cdot k) + k \cdot w(n/k)$, with the base condition $w(n) = O(n^2)$, for $n \leq k$, yielding $w(n) = O(n \cdot k \cdot \log_k n) = O(k \cdot \log_k n) \cdot w(MST(\vartheta_n))$.

Next, we show that the comparable monotone diameter $\bar{\Lambda}(n)$ of $H_k(n)$ is at most $O(\log_k n + \alpha(k))$. 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. Observe that $R(n)$ satisfies the recurrence $R(n) \leq 2 + R(n/k)$, with the base condition $R(n) = 1$, for $n \leq k$, yielding $R(n) = O(\log_k n)$. It is easy to verify 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 $\bar{\Lambda}(n) = O(\alpha(n))$, for $n \leq k$, yielding $\bar{\Lambda}(n) = O(\log_k n + \alpha(k))$.

Theorem 2.1 *For any n-point 1-dimensional Euclidean space and a positive integer k, there exists a 1-spanner with $O(n)$ edges, maximum degree at most $k+2$, diameter $O(\log_k n + \alpha(k))$ and lightness $O(k \cdot \log_k n)$.*

[5] In the 1-dimensional case any two points are comparable.

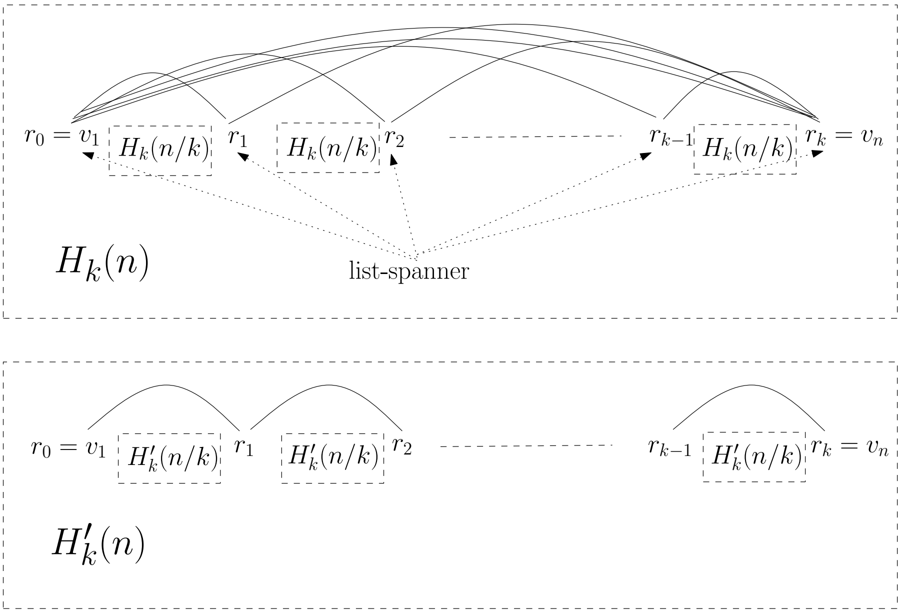

Figure 2: The constructions $H_k(n)$ and $H'_k(n)$ for a general k. Only the first level of the recursion is illustrated. (Edges of the original path P_n are not depicted in the figure.) For $H_k(n)$, all the cut-vertices are connected via the list-spanner, and, in addition, each of the two sentinels is connected to all other k cut-vertices. For $H'_k(n)$, each cut-vertex r_{i-1} is connected to the next cut-vertex r_i in line, $i \in [k]$.

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

We 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 $\Delta'(n) \le \max\{2, \Delta'(n/k)\}$, with the base condition $\Delta'(k) = 0$, for $n \le k$, yielding $\Delta'(n) \le 2$. Including edges of P_n, the maximum degree increases by at most two units, and so $\Delta(H'_k(n)) \le 4$. Consequently, the number of edges in $H'_k(n)$ is no greater than $2n$.

To bound the weight $w'(n) = w(H'_k(n))$ of $H'_k(n)$, first note that the path connecting all $k+1$ cut-vertices has weight $n-1$. Observe also that $w(P_n) = n-1$. Thus $w'(n)$ satisfies the recurrence $w'(n) \le 2(n-1) + k \cdot w'(n/k)$, with the base condition $w'(n) \le n-1$, for $n \le k$, yielding $w'(n) = O(n \cdot \log_k n) = O(\log_k n) \cdot w(MST(\vartheta_n))$.

Finally, the monotone radius $R'(n)$ of $H'_k(n)$ satisfies the recurrence $R'(n) \le k + R'(n/k)$, with the base condition $R'(n) \le n-1$, for $n \le k$, yielding $R'(n) = O(k \cdot \log_k n)$. It is easy to verify that the comparable monotone diameter $\bar{\Lambda}'(n) = \bar{\Lambda}'(H'_k(n))$ of the spanner $H'_k(n)$ satisfies the recurrence $\bar{\Lambda}'(n) \le \max\{\bar{\Lambda}'(n/k), k + 2R'(n/k)\}$, with the base condition $\bar{\Lambda}'(n) \le n-1$, for $n \le k$, yielding $\bar{\Lambda}'(n) = O(k \cdot \log_k n)$.

Theorem 2.2 *For any n-point 1-dimensional Euclidean space and a positive integer k, there exists a 1-spanner with maximum degree 4, diameter $O(k \cdot \log_k n)$ and lightness $O(\log_k n)$.*

2.3 General Tree Metrics

In this section we extend the constructions of Section 2.2 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 decomposes it into subtrees of size $O(|T|/k)$ each.

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 $c_1(v), c_2(v), \ldots, c_{ch(v)}(v)$. Suppose without loss of generality that the size of the left-most subtree $T_{c_1(v)}$ of v is no smaller than the size of any other subtree of v, i.e., $|T_{c_1(v)}| \geq |T_{c_2(v)}|, |T_{c_3(v)}|, \ldots, |T_{c_{ch(v)}(v)}|$. 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 $P(v) = (v, c_1(v), \ldots, l(v))$ the path of left-most edges leading down from v to the left-most vertex $l(v)$ in the subtree T_v of T rooted at v. A vertex v in T is called *d-balanced*, or simply *balanced* if d is clear from the context, if $|T_{c_1(v)}| \leq |T| - d$. The first balanced vertex along $P(v)$ is denoted by $b(v)$. For convenience, we write T_i (respectively, n_i) as a shortcut for $T_{c_i(b)}$ (resp., $|T_{c_i(b)}|$), for each $i \in [ch(b)]$.

Next, we present the Procedure CV that accepts as input a rooted tree (T, rt) and a parameter d, and returns as output a subset of $V(T)$. If $|T| < 2d$, the procedure returns the empty set $\emptyset$. Otherwise, for each child $c_i(b)$ of the first balanced vertex $b = b(rt)$ along $P(rt)$, $i \in [ch(b)]$, the procedure recursively constructs the subset $C_i = CV((T_i, c_i(b)), d)$, and then returns the vertex set $\bigcup_{i=1}^{ch(b)} C_i \cup \{b\}$. (See Figure 3 for an illustration.)

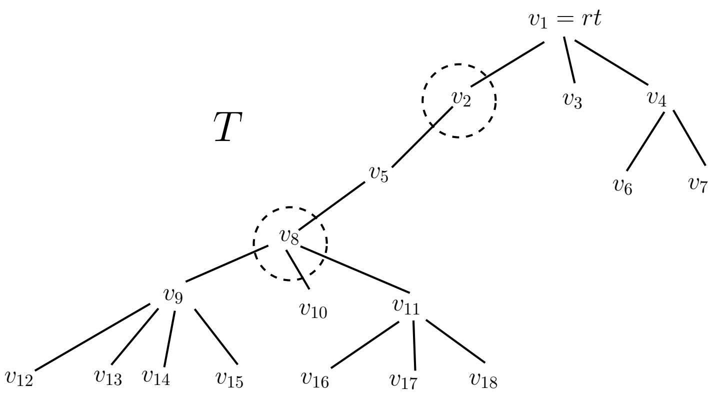

Figure 3: A rooted tree (T, rt) with $n = 18$ vertices $v_1 = rt, v_2, \ldots, v_{18}$. The first 6-balanced vertex along $P(rt)$ is v_2. The procedure CV on input (T, rt) and $d = 6$ returns the subset $\{v_2, v_8\}$.

Let (T, rt) be an n-vertex rooted tree, and let d be a fixed parameter. Next we analyze the properties of the subset $C = CV((T, rt), d)$ of *cut-vertices*.

Observe that for $n < 2d$, $C = \emptyset$, and for $n \geq 2d$, C is non-empty.
Next, we provide an upper bound on $|C|$ in the case $n \geq 2d$.

Lemma 2.3 *For $n \geq 2d$, $|C| \leq (n/d) - 1$.*

Proof: The proof is by induction on $n = |T|$.
Basis: $n \in [2d, 3d-1]$. 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 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 induction hypothesis, for each $i \in I$, $|C_i| \leq (n_i/d) - 1$. By construction, the vertex sets $C_1, C_2, \ldots, C_{ch(b)}$ and $\{b\}$ are pairwise disjoint and $C \ = \ \bigcup_{i=1}^{ch(b)} C_i \cup \{b\}$. Hence

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

The analysis splits into three cases depending on the size of $|I|$.
Case 1: $|I| = 0$. Equation (1) yields $|C| \leq 1 \leq (n/d) - 1$.
Case 2: $|I| = 1$. By construction, $n_1 \geq n_i$, for each $i \in [ch(b)]$, implying that $I = \{1\}$. Since b is balanced, $n_1 \leq n - d$, and so (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 (1) yields

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

$\square$

For a subset U of $V(T)$, 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 2.4 *The size of any tree in the forest $T \setminus C$ is smaller than $2d$.*

Proof: We start by proving the following claim.

Claim 2.5 *Let $\tilde{T}_b$ be the tree obtained from T by removing the subtree T_b. Then $|\tilde{T}_b| < d$.*

Proof: If $b = rt$, then $\tilde{T}_b$ is empty and the result is immediate. Otherwise, consider the parent $\pi(b)$ of b in T. Since b is the first balanced vertex along $P(rt)$, $\pi(b)$ is non-balanced, and so $|T_b| = |T_{c_1(\pi(b))}| > n - d$. Hence $|\tilde{T}_b| = n - |T_b| < d$, which completes the proof of Claim 2.5. $\square$
The proof of the lemma proceeds 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 and prove it for n. In this case, C is non-empty, and so the size of any tree in the forest $T \setminus C$ is strictly smaller than n. Consider a tree T' in the forest $T \setminus C$. Observe that for $n \geq 2d$,

$$T \setminus C \ = \ \bigcup_{i=1}^{ch(b)} (T_i \setminus C_i) \cup \{\tilde{T}_b\}.$$

Consequently, either $T' = \tilde{T}_b$, or it belongs to the forest $T_i \setminus C_i$, for some index $i \in [ch(b)]$. In the former case, the size bound follows from Claim 2.5, whereas in the latter case it follows from the induction hypothesis. Lemma 2.4 follows. $\square$

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 $\mathcal{Q} = \mathcal{Q}(T,C)$, and observe that for $n < 2d$, $\mathcal{Q} = \emptyset$.

Lemma 2.6 *For $n \geq 2d$, $\mathcal{Q}$ is a spanning tree of C rooted at $b = b(rt)$, with $\Delta(Q) \leq \Delta(T)$.*

Proof: We prove by induction on $n = |T|, n \geq 2d$, the following stronger statement: $\mathcal{Q}$ is a spanning tree of C rooted at $b = b(rt)$, such that for each vertex v in C, the number of children $ch_{\mathcal{Q}}(v)$ of v in $\mathcal{Q}$ is no greater than the corresponding number $ch(v)$ in T.
Basis: $n \in [2d, 3d-1]$. 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 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 $I = \{i_1, i_2, \ldots, i_I\}$. Observe that for each index $i \in [ch(b)] \setminus I$, $C_i = \emptyset$, and so $\mathcal{Q}(T_i, C_i)$ is an empty tree. By the induction hypothesis, for each $i \in I$, $\mathcal{Q}_i = \mathcal{Q}(T_i, C_i)$ is a spanning tree of C_i rooted at $b_i = b(c_i(b))$ in which the number of children of each vertex is no greater than the corresponding number in T_i. By construction, the only children of b in $\mathcal{Q}$ are the roots $b_{i_1}, b_{i_2}, \ldots, b_{i_I}$ of the non-empty trees $\mathcal{Q}_{i_1}, \mathcal{Q}_{i_2}, \ldots, \mathcal{Q}_{i_I}$, 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 tree in the forest $T \setminus C$ is incident to at most two cut-vertices. The proof of this lemma follows similar lines as those in the proof of Lemma 2.4, and is thus omitted.

Lemma 2.7 *For any tree T' in $T \setminus C$, no other vertex in T' other than its two sentinels is incident to a vertex in C. Moreover, $rt(T')$ is incident only to its parent in T, unless it is the root of T, and $l(T')$ is incident only to its left-most child in T, unless it is a leaf in T.*

Intuitively, Lemma 2.7 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. (See Figure 4 for an illustration.)

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.*

Lemmas 2.3, 2.4, 2.6 and 2.7 imply the following corollary, which summarizes the properties of the decomposition procedure.

Corollary 2.8 *1) For $n \geq 2d$, $|\tilde{C}| \leq (n/d) + 1$. 2) The size of any tree 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 tree T' in the forest $T \setminus \tilde{C}$, only the two sentinels of T' are incident to a vertex in $\tilde{C}$. Moreover, $rt(T')$ is incident only to its parent in T and $l(T')$ is incident only to its left-most child in T, unless it is a leaf in T.*

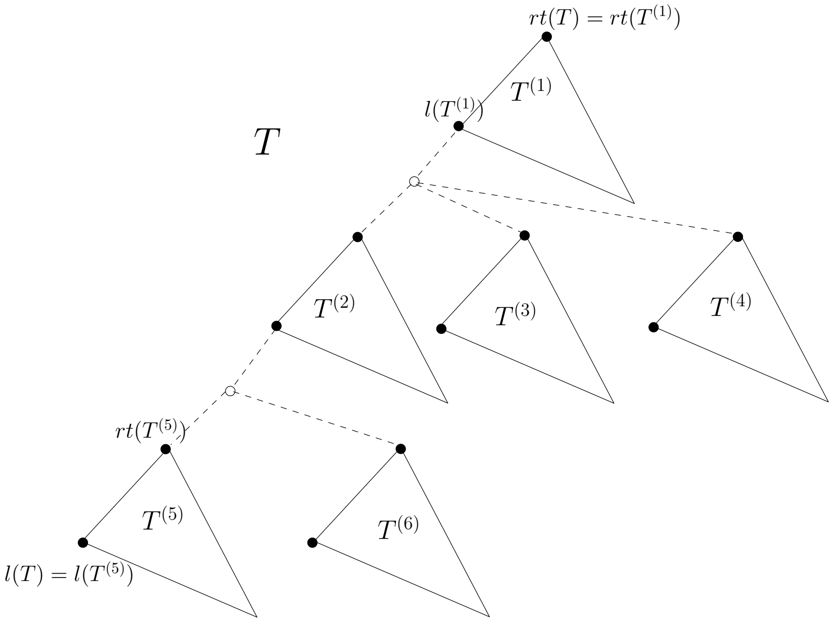

Figure 4: A "path-like" decomposition of the tree T into subtrees $T^{(1)}, T^{(2)}, \ldots, T^{(6)}$. The two cut-vertices in the figure are depicted by white dots, whereas the twelve sentinels of the subtrees $T^{(1)}, T^{(2)}, \ldots, T^{(6)}$ are depicted by black dots. Similarly to the 1-dimensional 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 construction 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 $4 \le k \le n/2 - 1$, and set $d = n/k$. 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 $k \ge 4$, it holds that $2d = 2n/k < n$. Denote the trees in the forest $T \setminus \tilde{C}$ by $\tilde{T}^{(1)}, \tilde{T}^{(2)}, \ldots, \tilde{T}^{(p)}$. By Corollary 2.8, $|\tilde{C}| \le k + 1$, and each tree $\tilde{T}^{(i)}$ in $T \setminus \tilde{C}$ has size less than $2n/k$. Observe that $\sum_{i=1}^{p} |\tilde{T}^{(i)}| = n - |\tilde{C}| \ge n - k - 1$, implying that $p \ge (n - k - 1)/(2n/k) \ge k/4$. (The last inequality holds for $k \le n/2 - 1$.)

To connect the set $\tilde{C}$ of cut-vertices, the algorithm first constructs the tree $\tilde{\mathcal{Q}} = \mathcal{Q}(T, \tilde{C})$. Observe that $\tilde{\mathcal{Q}}$ 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 $\tilde{\mathcal{Q}}$ if and only if it is its ancestor in T. Consequently, any 1-spanner path in $\tilde{\mathcal{Q}}$ between two arbitrary comparable[6] points is also a 1-spanner path for them in the original tree T. The algorithm proceeds by building a 1-spanner for $\tilde{\mathcal{Q}}$ via one of the aforementioned generalized constructions from [8, 2, 6, 33] (henceforth, *tree-spanner*). In other words, $O(k)$ edges between cut-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 is $O(\alpha(k))$. Then the algorithm adds to the

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

spanner $\mathcal{H}_k(n)$ edges that connect each of the two sentinels to all other cut-vertices. (In fact, the leaf $l(T)$ needs 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 $\tilde{T}^{(1)}, \tilde{T}^{(2)}, \ldots, \tilde{T}^{(p)}$ 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 both sentinels $rt(T)$ and $l(T)$ to all the other points.

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

$$E(n) \;\le\; c\cdot k + 4c\cdot \sum_{i=1}^{p} (|\tilde{T}^{(i)}| - 1) \;=\; c\cdot k - 4c\cdot p + 4c\cdot \sum_{i=1}^{p} |\tilde{T}^{(i)}| \;\le\; c\cdot (k-4p) + 4c\cdot (n-1) \;\le\; 4c(n-1).$$

(The last inequality holds as $p \ge k/4$.)

Denote by $\Delta(n)$ the maximum degree of a vertex in $\mathcal{H}_k(n)$, excluding edges of T. Since $|\tilde{C}| \le k+1$, $\Delta(n)$ satisfies the recurrence $\Delta(n) \le \max\{k, \Delta(2n/k)\}$, with the base condition $\Delta(n) \le 2k$, for $n < 2k+2$, yielding $\Delta(n) \le 2k$. Including edges of the tree T, the number of edges increases by $n-1$ units and the maximum degree increases by at most $\Delta(T)$ units.

Next, we show that the lightness $\Psi(\mathcal{H}_k(n))$ of the spanner $\mathcal{H}_k(n))$ satisfies $\Psi(\mathcal{H}_k(n)) = O(k \cdot \log_k n)$. To this end, we extend the notion of *load* defined in [1][7] for 1-dimensional spaces to general tree metrics. Consider an edge $e' = (v, w)$ connecting two arbitrary points in M_T, and a tree-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 a tree-edge e is called the load of e by H and it is denoted $\chi(e) = \chi_H(e)$. The load of H, $\chi(H)$, is the maximum load of a tree-edge by H. By double counting,

$$w(H) \;=\; \sum_{e' \in E(H)} w(e') \;=\; \sum_{e \in E(T)} \chi_H(e)\cdot w(e) \;\le\; \chi(H)\cdot \sum_{e \in E(T)} w(e) \;=\; \chi(H)\cdot w(T), \tag{2}$$

implying that $\Psi(H) = w(H)/w(T) \le \chi(H)$. Thus it suffices to provide an upper bound of $O(k \cdot \log_k n)$ on the load $\chi(n) = \chi(\mathcal{H}_k(n))$ of $\mathcal{H}_k(n)$. Since $\mathcal{H}_k(n)$ contains only $O(k)$ edges that connect cut-vertices, each subtree in the forest $T \setminus \tilde{C}$ is loaded by at most $O(k)$ such edges. In addition, $\mathcal{H}_k(n)$ contains all edges of the original tree T. These edges contribute an additional unit of load to each subtree in the forest $T \setminus \tilde{C}$. Hence $\chi(n)$ satisfies the recurrence $\chi(n) \le O(k) + \chi(2n/k)$, with the base condition $\chi(n) = O(n)$, for $n < 2k+2$, yielding $\chi(n) = O(k \cdot \log_k n)$.

Next, we show that $\bar{\Lambda}(n) = \bar{\Lambda}(\mathcal{H}_k(n)) = O(\log_k n + \alpha(k))$. 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 2.8, similarly to the 1-dimensional case, $\hat{R}(n)$ satisfies the recurrence $\hat{R}(n) \le 2 + \hat{R}(2n/k)$, with the base condition $\hat{R}(n) = 1$, for $n < 2k+2$, yielding $\hat{R}(n) = O(\log_k n)$. 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 $\check{R}(n) =$

[7] Agarwal et al. [4] used a slightly different notion which they called *covering*. The notion of load as defined above was introduced in [15], but the two notions are very close.

$O(\log_k n)$. Applying again Corollary 2.8 and reasoning similar to the 1-dimensional case, we get that $\bar{\Lambda}(n) \leq \max\{\bar{\Lambda}(2n/k), O(\alpha(k)) + \check{R}(2n/k) + \hat{R}(2n/k)\}$, with the base condition $\bar{\Lambda}(n) = O(\alpha(n))$, for $n < 2k+2$, yielding $\bar{\Lambda}(n) = O(\log_k n + \alpha(k))$.

Theorem 2.9 *For any tree-metric M_T and a positive integer k, there exists a 1-spanner with $O(n)$ edges, maximum degree at most $\Delta(T) + 2k$, diameter $O(\log_k n + \alpha(k))$ and lightness $O(k \cdot \log_k n)$.*

We remark that the maximum degree $\Delta(\mathcal{H})$ of the spanner $\mathcal{H} = \mathcal{H}_k(n)$ cannot be smaller than the maximum degree $\Delta(T)$ of the original tree in general. Indeed, consider a unit weight star T with edge set $\{(rt, v_1), (rt, v_2), \ldots, (rt, v_{n-1})\}$. 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 $\Omega(\log n)$.

The algorithm starts with constructing the tree $\tilde{\mathcal{Q}} = \mathcal{Q}(T, \tilde{C})$ that spans the set $\tilde{C}$ of cut-vertices. Observe that the depth of $\tilde{\mathcal{Q}}$ is at most k, implying that any two comparable cut-vertices are connected via a 1-spanner path in $\tilde{\mathcal{Q}}$ that consists of at most k hops. Since $\tilde{\mathcal{Q}}$ inherits the tree structure of T, this path is also a 1-spanner path in T. Then the algorithm calls itself recursively for each of the subtrees $\tilde{T}^{(1)}, \tilde{T}^{(2)}, \ldots, \tilde{T}^{(p)}$ of T. At the bottom level of the recursion, i.e., when $n < 2k + 2$, the algorithm adds no additional edges to the spanner.

Similarly to Section 2.3.2 we get that the number of edges in $\mathcal{H}'_k(n)$ is $O(n)$. Denote by $\Delta'(n)$ the maximum degree of a vertex in $\mathcal{H}'_k(n)$, excluding edges of T. By the third assertion of Corollary 2.8, $\Delta(\tilde{\mathcal{Q}}) \leq \Delta(T)$, and so $\Delta'(n)$ satisfies the recurrence $\Delta'(n) \leq \max\{\Delta(T), \Delta'(2n/k)\}$, with the base condition $\Delta'(n) = 0$, for $n < 2k + 2$, yielding $\Delta'(n) \leq \Delta(T)$. It follows that the maximum degree $\Delta(\mathcal{H}'_k(n))$ of $\mathcal{H}'_k(n)$ is at most $2 \cdot \Delta(T)$.

Next, we show that $\chi'(n) = \chi(\mathcal{H}'_k(n)) = O(\log_k n)$, which, by (2), implies that $\Psi(\mathcal{H}'_k(n)) = O(\log_k n)$. Observe that each tree $\tilde{T}^{(i)}$ in the forest $T \backslash \tilde{C}$ is loaded by at most one edge in $\tilde{\mathcal{Q}}$, namely, the edge connecting the parent of $rt(\tilde{T}^{(i)})$ in T and the left-most child of $l(\tilde{T}^{(i)})$ in T, if exists. In addition, $\mathcal{H}'_k(n)$ contains all edges of the original tree T. These edges contribute an additional unit of load to each subtree in $T \setminus \tilde{C}$. Hence $\chi'(n)$ satisfies the recurrence $\chi'(n) \leq 2 + \chi'(2n/k)$, with the base condition $\chi'(n) \leq 1$, for $n < 2k + 2$, yielding $\chi'(n) = O(\log_k n)$.

By Corollary 2.8, 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 $\hat{R}'(n) \leq n - 1$, for $n < 2k + 2$, 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 2.8 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 following recurrence $\bar{\Lambda}'(n) \leq \max\{\bar{\Lambda}'(2n/k), k + \check{R}'(2n/k) + \hat{R}'(2n/k)\}$, with the base condition $\bar{\Lambda}'(n) \leq n - 1$, for $n < 2k + 2$, yielding $\bar{\Lambda}'(n) = O(k \cdot \log_k n)$.

Theorem 2.10 *For any tree-metric M_T and a positive integer k, there exists a 1-spanner with $O(n)$ edges, maximum degree at most $2 \cdot \Delta(T)$, diameter $O(k \cdot \log_k n)$ and lightness $O(\log_k n)$.*

3 Euclidean Spanners

In this section we plug the 1-spanners for tree metrics from Section 2.3 on top of the dumbbell trees of [4] to obtain our constructions of Euclidean spanners.

Theorem 3.1 *("Dumbbell Theorem", Theorem 2 [4]) Given a set S of n points in $\mathbb{R}^d$ and a parameter $\epsilon > 0$, there exists a forest $\mathcal{D}$ consisting of $O(1)$ rooted binary trees, 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) Given any two points $u, v \in S$, there is a tree in $\mathcal{D}$, so that the path formed by walking from representative to representative along the unique path in that tree between these vertices, 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 to 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. (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 weight of dumbbell trees is always $O(\log n) \cdot w(MST(S))$, 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 remark 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 $\mathcal{H}^i = \mathcal{H}^i_k(O(n))$ of Theorem 2.9 for each of the tree metrics M_i. Then we translate each $\mathcal{H}^i$ to be a spanning subgraph $\breve{\mathcal{H}}^i$ of S in the obvious 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 $\breve{\mathcal{H}}^i$.

It is easy to see that the number of edges in $\mathcal{E}_k(n)$ is $O(n)$.

Next, we show that $\Lambda(\mathcal{E}_k(n)) = O(\log_k n + \alpha(k))$. By the Dumbbell Theorem, for any pair of points u and v in S, there exists 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 2.9 implies that there is a 1-spanner path P in $\mathcal{H}^i$ between u and v that consists of at most $O(\log_k n + \alpha(k))$ hops. By the triangle inequality, the weight of the corresponding translated path $\breve{P}$ in $\breve{\mathcal{H}}^i$ is no greater than the weight of $P_{u,v}$. Hence, $\breve{P}$ is a $(1+\epsilon)$-spanner path for u and v that consists of at most $O(\log_k n + \alpha(k))$ hops.

We proceed by showing that $\Delta(\mathcal{E}_k(n)) = O(k)$. Since each dumbbell tree DT_i is binary, theorem 2.9 implies that $\Delta(\mathcal{H}^i) = O(k)$. Recall that each label is used at most twice in DT_i, and so $\Delta(\breve{\mathcal{H}}^i) \le 2 \cdot \Delta(\mathcal{H}^i) = O(k)$. The union of $O(1)$ such graphs will also have maximum degree $O(k)$.

Finally, we argue that $\Psi(\mathcal{E}_k(n)) = O(k \cdot \log_k n \cdot \log n)$. Consider a dumbbell tree DT_i, and recall that $w(T_i) = O(\log n) \cdot w(MST(S))$. By Theorem 2.9, the weight $w(\mathcal{H}^i)$ of $\mathcal{H}^i$ is at most $O(k \cdot \log_k n) \cdot w(DT_i)$. By the triangle inequality, the weight of each edge in $\breve{\mathcal{H}}^i$ is no greater than the corresponding weight in $\mathcal{H}^i$, implying that $w(\breve{\mathcal{H}}^i) \le w(\mathcal{H}^i) = O(k \cdot \log_k n \cdot \log n) \cdot w(MST(S))$. The union of $O(1)$ such graphs will also have weight $O(k \cdot \log_k n \cdot \log n) \cdot w(MST(S))$.

To obtain our construction of Euclidean spanners with diameter in the complementary range $\Lambda = \Omega(\log n)$, we use our 1-spanners for tree metrics from Theorem 2.10 instead of Theorem 2.9.

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 $O(\log_k n \cdot \log n)$.*

4 Acknowledgements

We are grateful to Sunil Arya and David Mount for helping us to understand their work, and for addressing our questions.

References

[1] P. K. Agarwal, Y. Wang, and P. Yin. Lower bound for sparse Euclidean spanners. In *Proc. of 16th SODA*, pages 670–671, 2005.

[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 & Computational Geometry*, 9:81–100, 1993.

[4] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. of 27th STOC*, pages 489–498, 1995.

[5] S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. *Algorithmica*, 17(1):33–54, 1997.

[6] H. L. Bodlaender, G. Tel, and N. Santoro. Trade-offs in non-reversing diameter. *Nord. J. Comput.*, 1(1):111–134, 1994.

[7] H. T.-H. Chan and A. Gupta. Small hop-diameter sparse spanners for doubling metrics. In *Proc. of 17th SODA*, pages 70–78, 2006.

[8] B. Chazelle. Computing on a free tree via complexity-preserving mappings. *Algorithmica*, 2:337–361, 1987.

[9] B. Chazelle and B. Rosenberg. The complexity of computing partial sums off-line. *Int. J. Comput. Geom. Appl.*, 1:33–45, 1991.

[10] L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proc. of 2nd SOCG*, pages 169–177, 1986.

[11] G. Das, P. J. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Proc. of 9th SOCG*, pages 53–62, 1993.

[12] G. Das and D. Joseph. Which triangulations approximate the complete graph? In *Proc. of the International Symp. on Optimal Algorithms, volume 401 of Lecture Notes in Computer Science*, pages 168–192, 1989.

[13] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. In *Proc. of 10th SOCG*, pages 132–139, 1994.

[14] G. Das, G. Narasimhan, and J. S. Salowe. A new way to weigh malnourished euclidean graphs. In *Proc. of 6th SODA*, pages 215–222, 1995.

[15] Y. Dinitz, M. Elkin, and S. Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Proc. of 49th FOCS*, pages 519–528, 2008.

[16] G. N. Frederickson. A data structure for dynamically maintaining rooted trees. In *Proc. of 4th SODA*, pages 175–184, 1993.

[17] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. Approximate distance oracles for geometric graphs. In *Proc. of 13th SODA*, pages 828–837, 2002.

[18] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. Approximate distance oracles for geometric spanners. *ACM Transactions on Algorithms*, 4(1), 2008.

[19] J. Gudmundsson, G. Narasimhan, and M. H. M. Smid. Fast pruning of geometric spanners. In *Proc. of 22nd STACS*, pages 508–520, 2005.

[20] Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. In *Proc. of 19th PODC*, pages 41–50, 2000.

[21] J. M. Keil. Approximating the complete euclidean graph. In *Proc. of 1st SWAT*, pages 208–213, 1988.

[22] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete & Computational Geometry*, 7:13–28, 1992.

[23] Y. Mansour and D. Peleg. An approximation algorithm for min-cost network design. *DIMACS Series in Discr. Math and TCS*, 53:97–106, 2000.

[24] J. S. B. Mitchell. A ptas for tsp with neighborhoods among fat regions in the plane. In *Proc. of 18th SODA*, pages 11–18, 2007.

[25] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

[26] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.

[27] M. Pătraşcu and E. D. Demaine. Tight bounds for the partial-sums problem. In *Proc. of 15th SODA*, pages 20–29, 2004.

[28] S. Rao and W. D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proc. of 30th STOC*, pages 540–550, 1998.

[29] J. Ruppert and R. Seidel. Approximating the d-dimensional complete Euclidean graph. In *Proc. of 3rd CCCG*, pages 207–210, 1991.

[30] R. E. Tarjan. Applications of path compression on balanced trees. *J. ACM*, 26(4):690–715, 1979.

[31] M. Thorup. On shortcutting digraphs. In *Proc. of 18th WG*, pages 205–211, 1992.

[32] M. Thorup. Shortcutting planar digraphs. *Combinatorics, Probability & Computing*, 4:287–315, 1995.

[33] M. Thorup. Parallel shortcutting of rooted trees. *J. Algorithms*, 23(1):139–159, 1997.

[34] A. C. Yao. Space-time tradeoff for answering range queries. In *Proc. of 14th STOC*, pages 128–136, 1982.