2015 IEEE 56th Annual Symposium on Foundations of Computer Science

A light metric spanner

Lee-Ad Gottlieb
Department of Computer Science and Mathematics
Ariel University
Ariel, Israel
leead@ariel.ac.il

Abstract

It has long been known that d-dimensional Euclidean point sets admit $(1+\varepsilon)$-stretch spanners with lightness $W_E = \varepsilon^{-O(d)}$, that is the total edge weight is at most W_E times the weight of the minimum spanning tree of the set [DHN93]. Whether or not a similar result holds for metric spaces with low doubling dimension has remained an open problem. In this paper, we resolve the question in the affirmative, and show that doubling spaces admit $(1+\varepsilon)$-stretch spanners with lightness $W_D = (\operatorname{ddim}/\varepsilon)^{O(\operatorname{ddim})}$.

Keywords

Spanners, metric spaces, doubling dimension.

I. Introduction

Let $G = (V_G, E_G)$ be a metric graph, where vertices V_G represent points of some metric set S, while the edge weights of E_G correspond to inter-point distances in S. A graph $R = (V_R, E_R)$ is a $(1+\varepsilon)$-stretch *spanner* of graph $G = (V_G, E_G)$ if R is a subgraph of G (specifically, $V_R = V_G$ and $E_R \subset E_G$), and also $d_R(u,v) \le (1+\varepsilon)d_G(u,v)$ for all $u, v \in G$. Here, $d_G(u,v)$ and $d_R(u,v)$ denote the shortest path distance between u and v in the graphs G and R, respectively.

Low-stretch spanners have been the subject of intensive and broad-range study over the past three decades. Research has focused on minimizing such properties as construction time, vertex degree, graph diameter, and total edge weight – as well as possible trade-offs between these quantities – in various settings such as planar graphs or Euclidean spaces [Vai91], [Sal91], [Soa94], [AMS94], [CK95], [AS97], [DN97], [AMS99], [ADM$^+$95], [DNS95], [GLN02], [BGM04], [DES08], [ES13], [DHN93], [Sol11]. Of particular interest is a remarkable result of the nineties, that d-dimensional Euclidean spaces admit $(1+\epsilon)$-stretch spanners with *lightness* $W_E = \epsilon^{-O(d)}$, meaning that the total spanner weight is at most a factor W_E times the weight of the minimum spanning tree (MST) of the set [DHN93].

The work of Gao et al. [GGN06] (building upon classical work by Callahan and Kosaraju [CK95]) considered low-stretch spanners in metric spaces of low doubling dimension – a strictly more general setting that Euclidean space. This immediately spawned a long and fruitful line of work, showing that results comparable to those of Euclidean space can be obtained for these spaces as well, for construction time, degree and diameter. [CG06], [GR08b], [GR08a], [Smi09], [GKK13], [CLNS15], [Sol14]. However, the proof of lightness for Euclidean spanners relied heavily on the properties specific to that space (in particular, the *leapfrog property*) and so its analysis does not carry over to doubling spaces. The best lightness bound known for spanners in these metrics was $\Omega(\log n)$ [Smi09], [ES13]. Our contribution is in proving the following theorem:

Theorem I.1. *Given a complete graph G representing a metric S, there exists a $(1+\varepsilon)$-stretch spanner R of G with weight $W_D \cdot w(\mathrm{MST}(G))$ where $W_D = (\operatorname{ddim}/\varepsilon)^{O(\operatorname{ddim})}$ and $0 < \varepsilon < \frac{1}{2}$. Spanner R can be constructed in time $(\operatorname{ddim}/\varepsilon)^{O(\operatorname{ddim})} n \log^2 n$.*

Hence we resolve the question in the affirmative. We first prove that graphs which admit spanning trees that are everywhere sparse also admit light low-stretch spanners (Theorem III.4). We then show that doubling spaces

DOI 10.1109/FOCS.2015.52

can be decomposed into sets that are everywhere sparse, and that the light spanners on these sparse sets can be joined together into a light low-stretch spanner for the original set (Section IV).

Related work.: Light spanners are known for only a handful of settings. These include planar graphs [ADD+93], [ACC+96], unit disk graphs [KPX08], graphs which are snowflakes of metrics [GS14], and graphs of bounded pathwidth [GH12] and bounded genus [DHM10].

Bartal et al. [BGK12] (building upon Talwar [Tal04]) noted that the ability to efficiently construct a light spanner for doubling spaces would imply a much faster polynomial-time approximation scheme (PTAS) for the traveling salesman problem (TSP) in this setting. This approach parallels the result of Rao and Smith [RS98], who noted that for Euclidean TSP the solution tour may be restricted to the edges of a light low-stretch spanner of the set, and thereby improved the runtime of the Euclidean PTAS due to Arora [Aro98]. Our light spanner implies a much faster PTAS for TSP in doubling spaces than was previously known, but the analysis is beyond the scope of this paper and will appear elsewhere.

A. Preliminaries and notation

Graph properties. Throughout this paper, we will make the simplifying assumption that the input graph G is a complete graph.

Let the weight of an edge e $(w(e))$ be its length. Let the weight of an edge set E be the sum of the weights of its edges, $w(E) = \sum_{e \in E} w(e)$. Similarly, the weight of any graph $G = (V_G, E_G)$ is the weight of its edge-set, $w(G) = w(E_G)$. It follows that the weight of a path is its path length. For a partial graph $R = (V_R, E_R) \subset G$ (meaning $V_R \subset V_G$, $E_R \subset E_G$), $\text{diam}_G(R) = \text{diam}_G(V_R)$ is the diameter of R under the distance function d_G; it is the maximum distance in G between a pair of vertices also in R.

Let $B(u,r) \subset V_G$ refer to the vertices of V_G contained in the closed ball centered at $u \in V_G$ with radius r. $B^*(u,r) \subset E_G$ is the edge set of the complete graph on $B(u,r)$. A graph G is said to be *s-sparse* if for every radius r and vertex $v \in V$, the weight of $B^*(v,r) \cap E_G$ is at most sr. (Note that this is related to the c-packed property of curves.)

Let $A(u, r_1, r_2)$ be the annulus that includes all points at distance from u in the range $[r_1, r_2]$, and $A^*(u, r_1, r_2)$ the edge-set on the complete graph on $A(u, r_1, r_2)$. Points within distance r_1 of u are within the hollow of the annulus, and not part of it.

Doubling dimension. For a point set S, let $\lambda = \lambda(S)$ be the smallest number such that every ball in S can be covered by λ balls of half the radius, where all balls are centered at points of S. Then λ is the *doubling constant* of S, and the *doubling dimension* of S is $\text{ddim} = \text{ddim}(S) = \log_2 \lambda$ [Ass83]. The dimension is often taken to be an integer by rounding up the real number. The following lemma states the well-known packing property of doubling spaces (see for example [KL04]).

Lemma I.2. *If S is a metric space and $C \subseteq S$ has minimum inter-point distance b, then* $|C| = \left(\frac{2\,\text{rad}(S)}{b}\right)^{O(\text{ddim})}$.

Point hierarchies. Similar to what was described in [GGN06], [KL04] (though going back to at least Kolmogorov [Pan13]), a subset of points $X \subseteq Y$ is an r-net of Y if it satisfies the following properties:

(i) Packing: For every $x, y \in X$, $d(x,y) \geq r$.
(ii) Covering: Every point $y \in Y$ is strictly within distance r of some point $x \in X$: $d(x,y) < r$.

The previous conditions require that the points of X be spaced out, yet nevertheless cover all points of Y. A point in X covering a point in Y is called a *parent* of the covered point; this definition allows for a point to have multiple parents, but we shall assign to the point a single parent arbitrarily. Two net-points x, y of some r-net are called *c-neighbors* if $d(x,y) < cr$. By the packing property of doubling spaces, a net-point x has $c^{O(\text{ddim})}$ c-neighbors.

A hierarchy for a point set S is composed of nested levels of r-nets, where each level i of the hierarchy is a 2^i-net of the level $i-1$ beneath it. Note that the distance from an i-level point to all its descendants is less than $\sum_{j=0}^{\infty} 2^{i-j} = 2 \cdot 2^i$. By scaling, we will assume throughout this paper that the minimum inter-point distance in all sets is at least 1. Then we have that top level $H := \lceil \log_2 \mathrm{diam}(S) \rceil$ contains a single point, and the 0-level contains all of S. (Below, we will find it convenient to allow for levels $i < 0$, and each of these levels contains all of S as well.) The bottom level is L. A full hierarchy for S can be constructed in time $2^{O(\mathrm{ddim})} n \log \Delta$, where Δ is the *aspect ratio* of S, the ratio between the largest and smallest inter-point distance in S [KL04]. A hierarchy can also be constructed in time $2^{O(\mathrm{ddim})} n \log n$, and in this case some isolated net-points are represented implicitly [HM06], [CG06]. We can also augment the hierarchy with c-neighbor lists for all hierarchical levels and points: It is known that there are only $c^{O(\mathrm{ddim})} n$ neighbor pairs, and that given the hierarchy they may all be discovered in $c^{O(\mathrm{ddim})} n$ time and space.

An *r-semi-net* is a net that satisfies the covering property but not the packing property. A *semi-hierarchy* is composed of levels of nested semi-nets.

Net-Respecting spanning trees. We will say that a spanning tree T is *net respecting (NR)* relative to a given hierarchy if for every edge e in T, both of its endpoints are i-level net-points for i satisfying $12 \cdot 2^i \le w(e) < 24 \cdot 2^{i+1}$. (Recall from above that we allow levels below $i = 0$.) The following lemma is adapted from Lemma 1.6 in [GKK13].

Lemma I.3. *Every spanning tree T can be converted into a net-respecting spanning tree T' that visits all points visited by T, such that $w(T') \le 2.5 w(T)$.*

Proof: For every edge $e = (x, y)$ in T do the following. Let x', y' be i-level net-point ancestors of x, y respectively, for the lowest i satisfying $12 \cdot 2^i \le d(x', y') < 24 \cdot 2^{i+1}$. Replace edge e with long edge $e' = (x', y')$, and also add the short edges (x, x') and (y, y'). The short edges have weight at most $2 \cdot 2^i$. By the triangle inequality, $w(e) \ge w(e') - 4 \cdot 2^i \ge 8 \cdot 2^i$, and so $\frac{w(e')}{w(e)} \le \frac{w(e')}{w(e') - 4 \cdot 2^i} \le \frac{3}{2}$. While edge e' is net-respecting, edges (x, x') and (y, y') may not be. The short edges are themselves replaced by the procedure above. This leads to a series of edge replacements, and if the replacement adds an edge already in the set, we will remove the duplicate. The total weight of all short edges added by the recursive procedure is a geometric series summing to less than $8 \cdot 2^i \le w(e)$. Then the total weight of all edges is at most $w(e)(1 + \frac{3}{2}) = 2.5 w(e)$. The result follows by summing over all edges in T. ■

Let MST^{NR} denote the net-respecting minimum spanning tree. The following lemma relates a local minimum spanning tree to a global net-respecting minimum spanning tree. It is immediate from Lemma 1.11 in [GKK13], which made a similar claim concerning TSP tours.

Lemma I.4. *Let S be a point set equipped with a hierarchy. Let u be any point of S, and $r > 0$ any value. Then*

(i) $w(\mathrm{MST}^{NR}(S) \cap B^*(u, r)) \le 14(w(\mathrm{MST}(B(u, r)))$

(ii) $w(\mathrm{MST}(B(u, r))) \le 2w(\mathrm{MST}^{NR}(S) \cap B^*(u, 4r)) + 2^{O(\mathrm{ddim})} r$.

Let S' consist of $B(u, 2^i)$ along with all hierarchical ancestors of these points up to level i. Then $S' \subset B(u, 3 \cdot 2^i)$. We recall that the removal of points from a set can increase the weight of its minimum spanning tree by at most a factor of 2, and so $\frac{1}{2} w(\mathrm{MST}(B(u, 2^i))) \le w(\mathrm{MST}(S')) \le 2w(\mathrm{MST}(B(u, 3 \cdot 2^i)))$. We conclude that

$$\frac{1}{28} w(\mathrm{MST}^{NR}(S') \cap B^*(u, 2^i)) \le w(\mathrm{MST}(S')) \le 4w(\mathrm{MST}^{NR}(S') \cap B^*(u, 12 \cdot 2^i)) + 2^{O(\mathrm{ddim})} r. \qquad (1)$$

Finally, we note that for any S, $w(\mathrm{MST}(S)) \le |S| \,\mathrm{diam}(S)$. Tighter bounds are in fact known for Euclidean and doubling spaces [Aro98], [Tal04], [Smi10], but are not necessary for our results.

II. Background: Hierarchical spanners

Before presenting our results, we review the basic hierarchical spanner introduced by Gao et al. [GGN06], along with some small modifications suited for our purposes. This spanner provides a theoretical framework for our approach, although our ultimate construction will be significantly more involved. The results presented in this section are mostly well-known.

A *complete hierarchical spanner* R for a graph G representing a metric point set S is derived from the hierarchy of S. For some constant c, for each level i we add an edge between every pair of i-level net-points with inter-point distance at most $c2^i$. More precisely, we add to E_R an edge between the two vertices in V_R that represent these points. Recall that the top level of the hierarchy is H, and the minimum inter-point distance in S in 1. For this construction we will assume that the bottom level of the hierarchy is $L := \lfloor \log(1/c) \rfloor$. Let $M = H - L + 1$ be the number of levels in the hierarchy for S. To construct the complete hierarchical spanner, it suffices to construct a hierarchy for S and compute all relevant c-neighbors, in total time $2^{O(\text{ddim})} n \log n + c^{O(\text{ddim})} n$ [GR08b], [GR08a].

Lemma II.1. *Any hierarchical spanner R for graph G on S which satisfies that all i-level net-point c-neighbor pairs have stretch 1 (for all i), is a $(1 + \frac{32}{c})$-stretch spanner for all of G, when $c \geq 24$.*

Proof: Consider any two net-points x, y in the bottom level L of the hierarchy of S. Let x', y' be their respective i-level ancestors, where i is the minimum value for which x', y' are c-neighbors We have $d_G(x', y') \leq c2^i$. To derive a lower-bound on $d_G(x', y')$, let x'', y'' be the respective $(i-1)$-level ancestors of x, y, and by assumption $d_G(x'', y'') > c2^{i-1}$. Applying the triangle inequality, and recalling that $d_G(x', x'') \leq 2^i$ and $d_G(y'', y) \leq 2^i$, we have $d_G(x', y') \geq -d_G(x', x'') + d_G(x'', y'') - d_G(y'', y') > c2^{i-1} - 2 \cdot 2^i = (c-4)2^{i-1}$. Since x', y' are c-neighbors, $d_R(x', y') = d_G(x', y')$. recalling that the distance from an i-level point to all its descendants is at most $2 \cdot 2^i$, and applying the triangle inequality, we have $\frac{d_R(x,y)}{d_G(x,y)} \leq \frac{d_R(x,x') + d_R(x',y') + d_R(y',y)}{-d_G(x,x') + d_G(x',y') - d_G(y',y)} < \frac{d_G(x',y') + 4 \cdot 2^i}{d_G(x',y') - 4 \cdot 2^i} < \frac{(c-4)2^{i-1} + 2^{i+2}}{(c-4)2^{i-1} - 2^{i+2}} = \frac{c+4}{c-12} = 1 + \frac{16}{c-12} \leq 1 + \frac{32}{c}$, where the final inequality follows from $c \geq 24$. ■

The stretch of this spanner is arbitrarily low, but it has poor lightness bounds: In the trivial case where the n points of S reside on the line at intervals of distance 1, it is easy to see that $w(E_R) = \Theta(n \log n) = \Theta(\log n \cdot w(\text{MST}(S))$. We may consider being more stingy with added edges by using the greedy algorithm of [DN97] (see also [GLN02]), and we call this construction the *greedy hierarchical spanner*: We build the spanner as before, but first sort all $c^{O(\text{ddim})} n$ edges in increasing order. We consider each edge in turn, and before adding an edge between c-neighbors, we check to see if the stretch between them on the current partial spanner exceeds some threshold – for example, it is at most $1 + b$ for some constant $0 < b \leq 1$ – and only add the edge if the condition is met. In Lemma II.2 below, we show that the greedy hierarchical spanner has $1 + O(\frac{1}{c} + b)$ stretch.

Lemma II.2. *A hierarchical spanner R for graph G on S, for which all i-level net-point c-neighbor pairs have $(1 + b)$-stretch (for all i) for any constant $b > 0$, is a $(1 + \frac{32}{c} + 2b)$-stretch spanner for all of G, when $c \geq 24$.*

Proof: The proof is similar to the proof of Lemma II.1. Consider any two net-points x, y in the bottom level L of the hierarchy of S. Let x', y' be their respective i-level ancestors, where i is the minimum value for which x', y' are c-neighbors: $d_G(x', y') \leq c2^i$. As above, $d_G(x', y') > (c-4)2^{i-1}$. Now noting that by construction $d_R(x', y') \leq (1+b) d_G(x', y')$, and applying the triangle inequality, we have $\frac{d_R(x,y)}{d_G(x,y)} \leq \frac{d_R(x,x') + d_R(x',y') + d_R(y',y)}{-d_G x, x' + d_G(x',y') - d_G(y',y)} < \frac{(1+b) d_G(x',y') + 2 \cdot 2^{i+1}}{d_G(x',y') - 2 \cdot 2^{i+1}} < \frac{(1+b)(c-4)2^{i-1} + 2^{i+2}}{(c-4)2^{i-1} - 2^{i+2}} = \frac{c+4}{c-12} + b\frac{c-4}{c-12} < 1 + \frac{32}{c} + 2b$. ■

In closing this section, we note that the stretch bounds of Lemmata II.1 and II.2 also hold for net-points of semi-hierarchies. Indeed, the proofs use only the covering property of nets, and not their packing property.

III. A LIGHT SPANNER FOR GRAPHS WITH SPARSE SPANNING TREES

In this section, we show that graphs with net-respecting sparse spanning trees admit light low-stretch spanners. Note that we do not require the graph itself to be sparse, merely that it admit a sparse spanning tree. In Section III-A, we will show that we can build a light spanner for the union of two low-stretch paths. In Section III-B, we will use this result to build a light spanner for all graphs with sparse net-respecting spanning trees.

A. A light spanner for pairs of close low-stretch paths

In this section we show that given a close pair $P, Q \subset G$ of low-stretch paths, we can compute a light low-stretch spanner for their union. We first need to define nets, hierarchies and bipartite spanners for paths.

Path nets and hierarchies. Let $P \subset G$ be a path consisting of some vertices and edges in G. An *r-path-net* for P is an r-net for P under the path distance function d_P. A *path-hierarchy* for P is a full hierarchy under d_P, with net points in level $i = \lceil \log_2 w(P) \rceil$ and lower.

It is easy to see that an r-path-net of P is an r-semi-net for the point of P under d_G: We observe that the distance function d_P is non-contractive with respect to d_G; $d_G(u, v) \leq d_P(u, v)$ for all $u, v \in V$. Hence, if a point is covered under d_P, it is covered under d_G as well. However, the packing property under d_P may not hold under d_G, so the path-net is only a semi-net for P under d_G. It follows as well that the path-hierarchy for P is a semi-hierarchy for P under d_G.

Bipartite path spanners. Let $G = (V_G, E_G)$ be a complete graph, and let $P = (V_P, E_P), Q = (V_Q, E_Q)$ be two paths in G with stretch at most $(1 + b_1)$ for some $0 < b_1 < 1$. Let each path be equipped with a net-hierarchy. A *complete bipartite hierarchical spanner* R for $P \cup Q$ with parameter c contains all vertices of $V_P \cup V_Q$ and path edges $E_P \cup E_Q$ as well as the set of all edges $E \subset E_G$ connecting i-level vertices of V_P and V_Q within distance $c2^i$ (for all i). A *greedy bipartite hierarchical spanner* R for $P \cup Q$ contains all vertices of $V_P \cup V_Q$ and path edges $E_P \cup E_Q$, as well as an edge-set $E' \subset E \subset E_G$: We sort the edges of E in increasing order, and consider each edge in turn. An edge of E is added to E' only if the stretch between its endpoints on the current partial spanner is large, say more than $(1 + b_1 + b_2)$ for some $0 < b_2 \leq 1 - b_1$. We show that the greedy bipartite spanner has favorable properties:

Lemma III.1. *Let $P, Q \subset G$ be a pair of $(1 + b_1)$-stretch paths whose distance from each other*[1] *is not greater than $c \cdot \min\{w(P), w(Q)\}$ (for $0 \leq b_1 < 1$ and any $c \geq 24$). Let Δ be the aspect ratio of $P \cup Q$. A greedy bipartite hierarchical spanner R with parameters c and $0 < b_2 \leq 1 - b_1$ for paths P, Q can be constructed in time $O(c(|V_P| + |V_Q|) \log(c\Delta))$, and satisfies the following properties:*

(i) *Stretch: $d_R(p, q) \leq (1 + \frac{32}{c} + 2(b_1 + b_2))d_G(p, q)$ for all $p \in P$ and $q \in Q$.*
(ii) *Weight: $w(E') = \frac{16c^2}{b_2} \cdot \min\{w(P), w(Q)\}$, where $E' = E_R - (E_P \cup E_Q)$ is the set of new spanner edges not in P or Q.*

Proof: **(i)** This follows immediately from Lemma II.2.

(ii) The proof is by a charging argument. We assume without loss of generality that $w(P) \leq w(Q)$. Take the maximum level i for which E (the edge set of the complete bipartite hierarchical spanner) has an i-level edge, and let the endpoints of the edge be $p \in P$ and $q \in Q$. Consider any point $\tilde{p} \in P$ satisfying $d_P(p, \tilde{p}) \leq r$ for $r = \frac{b_2}{4c} d_G(p, q)$, and we will show that $\tilde{p}$ cannot have *any* edge incident to Q in E_R:

First note that the proximity of net-point p to $\tilde{p}$ implies that $\tilde{p}$ may be a j-level net-point only for values $j \leq \log r$. Now suppose by way of contradiction that $\tilde{p} \in P$ has an edge to some $\tilde{q} \in Q$, and then by construction it must be that $d_G(\tilde{p}, \tilde{q}) \leq c2^j \leq cr$. Let R' be the partial spanner before the addition of the i-level edges. Then

[1]Here, the distance between two paths is the smallest inter-point distance between them.

the stretch guarantees of the paths and spanner, along with two applications of the triangle inequality, imply that

$$\begin{aligned}\frac{d_{R'}(p,q)}{d_G(p,q)} &\le \frac{d_{R'}(p,\tilde{p}) + d_{R'}(\tilde{p},\tilde{q}) + d_{R'}(\tilde{q},q)}{d_G(p,q)}\\ &\le \frac{r + cr + (1+b_1)d_G(\tilde{q},q)}{d_G(p,q)}\\ &\le \frac{r + cr + (1+b_1)[d_G(p,q) + d_G(p,\tilde{p}) + d_G(\tilde{p},\tilde{q})]}{d_G(p,q)}\\ &\le \frac{(2+b_1)(r+cr) + (1+b_1)d_G(p,q)}{d_G(p,q)}\\ &= 1 + b_1 + \frac{(2+b_1)(r+cr)}{d_G(p,q)}\\ &< 1 + b_1 + \frac{4cr}{d_G(p,q)}\\ &= 1 + b_1 + b_2.\end{aligned}$$

This implies that in R' the stretch from p to q is not sufficient to add to E' an edge connecting them – a contradiction.

It follows that for any i-level edge $e \in E'$ connecting $p \in P$ to $q \in Q$, the entire path of P within distance $\frac{b_2}{4c}w(e)$ of p under d_P has no lower level edges incident upon it. We charge the i-level edge to this segment of the path P.

The statement now follows by noting that p may have at most $4c$ i-level edges in E' incident upon it (all of which will be charged to the same path segment of P): Let $q_f, q_l \in Q$ be the first and last i-level net-points in path Q that are endpoints of edges incident upon p. Then $d_Q(q_f, q_l) \le (1+b_1)d_G(q_f, q_l) \le (1+b_1) \cdot 2c2^i < 4c2^i$. There can be at most $4c$ i-level path-net points on a path of this length.

Runtime. At each of $O(\log(c\Delta))$ levels i, we must locate for each i-level path-net point $p \in P$ its i-level path-net c-neighbors in Q. As shown above, there are most $3cs$ such neighbors. Further, if two points $p \in P$ and $q \in Q$ are i-level c-neighbors, then so are their respective parents $p' \in P$ and $q' \in Q$: $d_G(p', q') \le d_G(p, p') + d_G(p, q) + d_G(q, q') \le 4 \cdot 2^i + c2^i = (c+4)2^i < c2^{i+1}$. So to compute all c-neighbors, it suffices to iterate down the hierarchy, maintaining for each i-level net-point $p \in P$ a list of all c-neighbors in Q. The lists for the children of p can be found by considering all children of p's c-neighbors, for a total of $O(c)$ candidates. ■

B. Extension to graphs with sparse spanning trees

The results in the previous section apply only to pairs of low-stretch paths. We will use them to obtain similar results for all graphs with sparse spanning trees. Our plan is as follows: We first show how to *decompose* a spanning tree into paths without a stretch guarantee (Lemma III.2). We then show how to *replace* any path with a light set of low-stretch paths (Lemma III.3). Finally, we build the bipartite spanner of the previous section on all pairs of low-stretch paths, and show that this gives a light low-stretch spanner tree for the full set (Theorem III.4). To do this, we will need to demonstrate that the union of all the low-stretch paths is still relatively sparse, and also that the union of the path-nets for these paths can serve as a semi-hierarchy for the entire space.

Lemma III.2. *Given an s-sparse spanning tree $T = (V_T, E_T) \subset G$ of n nodes with aspect ratio Δ, T may be decomposed in time $O(s|V_T|\log\Delta)$ into a set $\mathcal{Q}$ of paths, with the following properties:*

(i) *Diameter: At least one path $P_i \in \mathcal{Q}$ has length* $\mathrm{diam}_T(T)$.

(ii) *Proximity to path-net points: Every vertex in V_G is within distance b (under d_T) of some path $P_j \in \mathcal{Q}$ of length $w(P_j) \ge b$, for all values* $0 < b \le \mathrm{diam}_T(T)$.

Proof: The decomposition procedure removes from the spanning tree the longest path – of length $\operatorname{diam}(T)$ – and places the path in the collection $\mathcal{Q}$. That is, the edges of the path are removed from E_T, and then all vertices with no edges incident upon them are removed from V_T. After the removal of the longest path, a number of disjoint subtrees may remain, and each is decomposed recursively in the same manner until V_T is empty. This completes the description of the decomposition procedure.

We note trivially that the distance under d_T from a removed path to all vertices in one of the remaining trees is not greater than the graph diameter of that tree.

(i) The first item follows by construction.

(ii) Take any vertex and the paths that were removed from the spanning tree containing the vertex at each iteration. The paths are necessarily in decreasing order of length. Consider the first path of length at most b. Then the diameter of the subtree that contained this path before its removal was at most b, hence the the vertex is within distance b of the previously removed path, which has length greater than b.

Runtime. First note trivially that the longest path of a tree can be found in $O(n)$ time. Now consider any subtree formed by the removal of a path P of length $w(P)$. After $O(s)$ further decomposition steps, the longest path in any remaining subtrees cannot be longer than $\frac{w(P)}{2}$. This is because each decomposition step removes the longest path of the current subtree, and leaves behind it a group of subtrees all within distance $w(P)$ of P. Then the fact that the original subtree is s-sparse implies that it may contain at most $O(s)$ paths of length greater than $\frac{w(P)}{2}$. It follows that the total runtime of the decomposition procedure is $O(s|V_P|\log\Delta)$. ■

Note that the sparsity condition of Lemma III.2 was used only in bounding the runtime.

The paths in $\mathcal{Q}$ may have arbitrary stretch, and so it remains to replace each path in $\mathcal{Q}$ with a set of low-stretch paths. We will show how to do this in the case that the original spanning tree was net-respecting, and so the decomposed paths are net-respecting as well.

Consider the following procedure (motivated by the greedy algorithm) which takes a path P and value c, and replaces P with a set of low-stretch paths $\mathcal{P}$. First consider the endpoints $p, q \in P$. If the endpoints are close together – $d_P(p,q) \le \frac{1}{3}\operatorname{diam}_G(P)$ – there must exist a point r satisfying $d_G(p,r), d_G(q,r) \ge \frac{1}{3}\operatorname{diam}_G(P)$. Cut P at r, and continue the procedure below separately on the two smaller paths. Otherwise, continue the procedure on P itself:

Iteratively create a path-net for P. Set L to be the minimum interpoint distance in P, divided by c. At the first iteration $i = L$, each point in P is assigned to be an i-level net-point. At each subsequent iteration $i = L+1, \ldots$ promote some of the $(i-1)$-level path-net points to also be i-level path-net points: We begin by promoting the first point in P – call this p – and then proceed down the path to promote in turn every $(i-1)$-level point at path distance 2^i or greater from the previously promoted point.

After each iteration i of the path-net construction, we begin with the endpoint p and consider all j-level path-net points (for $j = i - \lceil\log(12c)\rceil$) down the path from p, within path distance $5 \cdot 2^i$ of p. If we find that the stretch between any pair is $1 + \frac{1}{3c}$ or greater, we remove a partial path of length at least $10 \cdot 2^i$ (under d_G) containing all these points, and replace the removed path by a single edge. The removed partial path is then segmented into smaller paths at its j-level path-net points, and these small paths are all added to $\mathcal{P}$. The search then moves on to iteratively consider the next (surviving) i-level path-net point in path P, and terminates at the final i-level point. Then the tail past the i-level point is segmented and added to $\mathcal{P}$, as above.

It remains only to specify exactly what partial path is removed above. Let p be the i-level net-point under consideration, and t be the last j-level net-point within distance $5 \cdot 2^i$ of p for which large stretch was seen. Let t' be the first point past t satisfying $10 \cdot 2^i \le d_G(p,t') < 48 \cdot 25 \cdot 2^i$, and the removed partial path is the one connecting p, t'. If no such point t' exist, then let p' be the first point preceding p on the path satisfying $10 \cdot 2^i \le d_G(p',t) < 48 \cdot 25 \cdot 2^i$, and the removed partial path is the one connecting p', t. (Note that such a point p' must exist, since the path is net-preserving, and so p and t cannot both be incident on edges of length 48 times their distance).

Lemma III.3 below shows that the final path set $\mathcal{P}$ has favorable properties, which we will use in the construction of the spanner.

Lemma III.3. *Let $P = (V_P, E_P)$ be an s-sparse net-respecting path in G with arbitrary stretch and aspect ratio Δ. For any constant $c \geq 24$, the collection $\mathcal{P}$ for P may be computed in time $O(c^2|V_p|\log\Delta)$, and possesses the following properties:*

(i) *Vertex cover: The union of vertices in all sets $P_i \in \mathcal{P}$ is exactly V_P ($\cup_i V_{P_i} = V_P$), and $\sum_i |V_{P_i}| \leq 3|V_P|+1$.*

(ii) *Stretch: Each path in $\mathcal{P}$ has stretch at most $(1+\frac{34}{c})$.*

(iii) *Diameter: At least one path $P_i \in \mathcal{P}$ has length $\frac{\mathrm{diam}_G(P)}{4} \geq \frac{\mathrm{diam}_P(P)}{4s}$ or greater.*

(iv) *Proximity to path-net points: Every vertex $v \in V_P$ is within d_G distance $120cb$ of some path $P_j \in \mathcal{P}$ of weight $w(P_j) \geq b$, where b is any value satisfying $0 < b \leq \frac{\mathrm{diam}_G(P)}{9}$.*

(v) *Path sparsity: Each path in $\mathcal{P}$ is $2s$-sparse.*

(vi) *Graph sparsity and weight: The union of all paths $P_i \in \mathcal{P}$ form a graph that is both $2s(2c+1)$-sparse and has total weight at most $2s(2c+1) \cdot w(P)$.*

Proof:

(i) This covering is immediate from the construction, and the number of vertices follows by noting that a removed path may duplicate the path endpoints, but removes at least a single vertex from the original path. The additive 1 comes from the preliminary case where the path endpoints are close together, in which case the middle point r is doubled.

(ii) Let p, q be any pair of k-level c-neighbors in path $P' \in \mathcal{P}$. We show that the path stretch of p, q is less than $1 + \frac{1}{c}$. Then the item follows from Lemma II.2.

Assume by contradiction that the stretch is at least $1 + \frac{1}{c}$, while p, q are found on the same path. Let r be the i-level point preceding p (or possible p itself) for $2^i \leq d_P(p,q) < 2 \cdot 2^i$. Then both p, q and their respective closest j-level ancestors p', q' (for j as in the algorithm description above) are within distance $5 \cdot 2^i$ of r. Now these ancestors covering have distortion at least $\frac{d_P(p',q')}{d_G(p',q')} \geq \frac{d_P(p,q)-4\cdot 2^j}{d_G(p,q)+4\cdot 2^j} > \frac{d_P(p,q)-4\cdot 2^j}{(1-\frac{1}{c})d_P(p,q)+4\cdot 2^j} \geq \frac{1-\frac{1}{3c}}{1-\frac{2}{3c}} > 1 + \frac{1}{3c}$, so this path must have been segmented and split at its j-level net-points. Further, there must be a j-level net-point between p, q, contradicting the assumption that p, q are on the same path: If there is no j level net-point between p, q, then all points between p, q are within distance $2 \cdot 2^j$ of p or q, and two applications of the triangle inequality imply that $d_P(p,q) \leq d_G(p,q) + 4 \cdot 2^j \leq d_G(p,q) + \frac{d_P(p,q)}{3c}$, and so $d_P(p,q) < (1+\frac{1}{3c})d_G(p,q)$, again a contradiction.

(iii) The endpoints $p, q \in P$ were chosen so that $\mathrm{diam}_P(P) \geq \frac{\mathrm{diam}_G(P)}{3}$. By construction, the final path remaining after all iterations retains p as an endpoint. On the other end, at each iteration i an end-segment of the path may be segmented. Let k be the penultimate iteration (the last iteration at which the tail is removed) so that the removed path has length at most 2^k, while the path at this iteration has size at least $\frac{c}{2}$ times this. The removed tails sum to less than $\sum_{i=L}^{k} 2^i < 2 \cdot 2^k$, a length that is a fraction at most $\frac{4}{c} \leq \frac{1}{6}$ of the path. So the diameter is at least $\frac{\mathrm{diam}_G(P)}{3} \cdot \frac{5}{6} > \frac{\mathrm{diam}_G(P)}{4}$. Finally, the s-sparsity of path P directly implies that $\mathrm{diam}_P(P) \leq s\,\mathrm{diam}_G(P)$.

(iv) Every point of the removed path was all within distance $5 \cdot 2^i$ of both p and t, so it is within distance $5 \cdot 2^i$ of an edge of length at least $10 \cdot 2^i$.

Now the removed path may itself have had paths removed from it at earlier stages of the iteration, but all these paths must be within distance $\sum_{j=L}^{i} 10 \cdot 2^j < 20 \cdot 2^i$ of the new edge. Also note that when this path was removed, it was decomposed into smaller paths of path-length at least $2^j = \frac{2^i}{12c}$. Hence, the lengths of the edges feature jumps of at most $12c$, and so every vertex is within distance $120cb$ of an edge of length at least b.

(v) Take an edge added by the procedure between points p', t or p, t'. Then in the original path P there must have been a heavier path connecting p', t, and by the arguments in item (iv) this path was fully within distance $5 \cdot 2^i$ of both endpoints. Any ball containing the new edge must contain all the removed path with its radius and so it contains the entire (removed) old path within 1.5 times its radius. Earlier paths may have also been removed, but by the same argument they are all within distance 2 times the radius. So the weight of the smaller ball is less than the weight of the larger larger ball, and the sparsity is $2s$.

(vi) For the claim of weight: We have already noted in item (iii) that the edges added when removing an end-segment sum to much less than the weight of the path. For the other edges, the proof proceeds by a charging argument. For an edge e added between points p, q in iteration i, charge its weight to the edges of the removed path. Each edge e' in the removed partial path Q receives change $w(e)\frac{w(e')}{w(Q)}$, and since Q must have weight at least $w(e)(1+\frac{1}{2c})$, the charge to e' is at most $\frac{w(e')}{1+\frac{1}{2c}}$. Now, if e' itself was added in an earlier iteration j, then the new charge to e' due to the addition of e is itself passed down to earlier removed paths, until all charges lie solely on edges present in the original path P. Let e be an edge of P, and it follows that the sum of charges placed on e (including the cost of e itself) is less than $w(e)\sum_{i=0}^{\infty}(1+\frac{1}{2c})^{-i} = (2c+1)w(e)$.

For the claim of sparsity: It follows from the proof of items (iii) and (iv) that all edges of $\mathcal{P}$ inside an r-radius ball charge to edges of the original path P within distance $2r$ of the ball's center. As P is s-sparse, the total weight in the small ball is $2s(2c+1)$, and the sparsity follows.

Runtime. There are $O(\log \Delta)$ iterations, and at each one we can make a single pass on the path, promoting net-points and recording their order in a list. For each i-level net-point we must consider all j-level pairs, of which there are $O(c^2)$. The runtime follows. ■

Joining together Lemmata III.1, III.2 and III.3, we can prove that spaces with sparse spanning trees admit light low-stretch spanners:

Theorem III.4. *Let $T = (V_T, E_T)$ be an s-sparse spanning tree of a complete graph G with aspect ratio Δ. Then in time $(s/\varepsilon)^{O(1)} \cdot |V_T| \log \Delta$ we can construct for G a spanner R with*

(i) *Weight: $W_s \cdot w(T)$ for $W_s = (s/\varepsilon)^{O(1)}$.*

(ii) *Stretch: $(1+\varepsilon)$.*

Proof: The construction is straightforward: Decompose the spanning tree T into paths using the *decomposition* procedure of Lemma III.2, and then replace each path with a set of low-stretch paths using the *replacement* procedure of Lemma III.3 with parameter $c_1 = 8 \cdot 34/\varepsilon$. For every pair of paths in the new set, build the greedy bipartite spanner preceding Lemma III.1 with parameters $c_2 = 64f/\varepsilon$ (for some constant $f > 1$ to be chosen below) and $b_2 = \varepsilon/8$. The new graph is G'. Recall that the bipartite spanner will only add edges if the path pair is sufficiently close.

(i) We first must prove sparsity: Tree T is s-sparse, and the sparsity condition is not affected by the decomposition. Take any path P the decomposition, and P is replaced by a set of paths connecting the vertices of P with sparsity $O(c_1 s)$ (Lemma III.3(vi)). Hence, before the addition of the bipartite edges G' possessed sparsity $s' = O(c_1 s^2)$.

Returning to proof of weight, it follows from Lemma III.3(vi) that the weight of the paths of G' before the construction of the bipartite spanners was $2s(2c+1) \cdot w(T)$. A bipartite spanner with parameter c_2 is built for each pair of paths, and we will charge the cost of the new edges to the shorter path. By sparsity, any path P is within distance $c_2 \cdot w(P)$ of $O(c_2 s') = O(c_1 c_2 s^2)$ longer paths. Charge P for the additional edges added by the bipartite spanner to each of these paths. By Lemma III.1 this yields a charge of $O(c_2^2/b_2) \cdot w(P)$ per spanner, for a total charge of $O(c_1 c_2^3 s^2/b_2) \cdot w(P)$. It follows that $w(G') = O(c_1 c_2^3 s^2/b_2) \cdot w(T)$.

(ii) By Lemma III.3, the intra-path stretch in all paths produced by the replacement procedure is $1 + \frac{34}{c_1} = 1 + \frac{\varepsilon}{8}$. By construction of the bipartite spanner, the stretch among all i-level path-net points that are c_2-neighbors is $1 + b_2 = 1 + \frac{\varepsilon}{8}$. Below, we show that we can create a semi-hierarchy where all c_2-neighbors have stretch at most b_2. Then by Lemma II.2, the resulting spanner has stretch less than $1 + \frac{32}{64/\varepsilon} + 2(\frac{\varepsilon}{8} + \frac{\varepsilon}{8}) = 1 + \varepsilon$.

To show that there exists a semi-hierarchy: We note that as a consequence of Lemmatta III.2(ii) and III.3(iv) each vertex v in G' is within distance $O(2^i/\varepsilon)$ of some i-level path-net point for all levels $i \leq \lceil \mathrm{diam}_T(T)/8s \rceil$. It follows that if we promote all i level path-net points to an appropriate level, we can ensure that each vertex v is within distance at most 2^j of some j-level point, and this hold for all levels $j \leq \lceil \mathrm{diam}_T(T) \rceil = H$. Hence we have a valid semi-hierarchy for all points. The final result follows by choosing an appropriate constant f in the definition of c_2.

Runtime. By Lemmata III.2 and III.3, the decomposition and replacement steps can be done in time $O(c_2^2 |V_T| \log \Delta)$. To compute each bipartite spanners, for each path P in G' we must have all paths of length $w(P)$ or greater within distance $c_2 \cdot w(P)$ of P. Recall from the proof of **(i)** above that the sparsity of G' before the addition of bipartite spanner edges is $s' = (s/\varepsilon)^{O(1)}$.

After the initial decomposition step, assign a path-hierarchy to each path, and compute all i-level $2c_2$-neighbors among all paths and pairs of paths, for all i. This can all be done in time $(s/\varepsilon)^{O(1)} n \log \Delta$. (The construction and analysis is identical to that of Lemma III.2.) For each path e created by the replacement step, we compute its distance to all i-level path-net points in the original edge within distance $c_2 2^i$. Since the current graph is s'-sparse and has aspect ratio Δ this can be done in time $(s/\varepsilon)^{O(1)} n \log \Delta$ over all paths, if we precomputed for each vertex its closest i-level path-net point for all i. For each i-level path-net point, we maintain a list of the replacement path which discovered this point, sorted in order of path length. Note that we may sort all replacement paths once, and then compute close path-net point for each one in turn.

For each path P and each $i \leq \log(w(e))$, we inspect the i-level path-net points within distance $c_2 2^i$, and all i-level c_2-neighbors of those path net points, and their listed paths of length greater than $w(P)$, and this suffices to discover for P all longer edges within distance $c_2 \cdot w(P)$. The runtime follows from the sparsity and aspect ratio of the current graph. ■

IV. A LIGHT SPANNER FOR COMPLETE METRIC GRAPHS

In Section III we showed how to construct a light spanner for spaces that have sparse net-respecting spanning trees. In this section we complete the proof of Theorem I.1 by showing how to decompose general metric graphs of low doubling dimension into graphs with sparse spanning trees. Then the light spanners for the decomposed sparse spaces can all be joined into a single light spanner for the original metric space. Our approach uses a technique developed for computing near-optimal traveling salesman tours in polynomial time [BGK12], [BG13], although our setting is much less restrictive, and so the problem of finding a good decomposition is simpler.

For some graph G equipped with a hierarchy, define $F(u,i)$ for i-level vertex u to include all points of $B(u, 2^i) \cap G$, as well as all their hierarchical ancestors up to level i. Note that $F(u,i) \subset B(u, 3cdot 2^i)$. The following preliminary lemma shows that we can spin off from G a (slightly) dense area of the graph D, and this deletes the dense area from G:

Lemma IV.1. *Let G be a graph with aspect ratio Δ, equipped with a hierarchy. For some fixed $f = (\mathrm{ddim}/\varepsilon)^{O(\mathrm{ddim})}$, let i be the lowest level for which there exists some u satisfying $\mathrm{MST}^{NR}(F(u,i)) > f2^i$. Then G may be segmented into two intersecting subgraphs $G', D \subset G$ with the following properties, for any value $c \geq 2$:*

- (i) *Weight in G': $w(\mathrm{MST}^{NR}(F(u,i) \cap G')) \leq \frac{f}{4} 2^i$, while $w(\mathrm{MST}^{NR}(G)) - w(\mathrm{MST}^{NR}(G')) \geq \frac{f}{8} 2^i$.*
- (ii) *Sparsity of D: $\mathrm{MST}^{NR}(D)$ is $c^{O(\mathrm{ddim})} f$-sparse, while D has diameter $O(c) \cdot 2^i$.*
- (iii) *Neighbor proximity: For every c-neighbor point pair in the hierarchy of G, the pair is found together in D or G' or in both.*

Proof: Subset D includes all points in $B(u, (13+c) \cdot 2^i)$, along with all their net-points up to level i. Subset G' includes all points of $G - B(u, 13 \cdot 2^i)$, as well as all i-level (and higher level) net-points in $B(u, 13 \cdot 2^i)$. Below, we will also add some more points to G'.

(ii) Recall that by assumption each $(i-1)$-level point v satisfies $\mathrm{MST}^{NR}(F(v, i-1)) \leq f2^{i-1}$. A spanning tree for any ball of radius r in D can be formed by covering that ball with the spanning trees of $r^{O(\mathrm{ddim})}$ sets $F(v, i-1)$, and then connecting the centers of these sets at an additional cost of $r^{O(\mathrm{ddim})} 2^i$, for a total weight of $r^{O(\mathrm{ddim})} f 2^i$.

(iii) Clearly c-neighbor pairs of levels i or higher are both found in G'. For levels $k < i$, if one of the points is not found in G', then by construction it must be found in $B(u, 13 \cdot 2^i)$, and so the second point is found in $B(u, 13 \cdot 2^i + c2^k) \subset B(u, (13+c) \cdot 2^i) = D$.

(i) We first discuss the weight of $\mathrm{MST}^{NR}(G)$ in the area around u: A consequence of Equation (1) is that $B^*(u, 12 \cdot 2^i)$ had intersected edges of $\mathrm{MST}^{NR}(G)$ weighing more than $\frac{f}{4} 2^i$. At the same time, the larger ball $B(u, 13 \cdot 2^i)$ admitted a minimum spanning tree of weight $2^{O(\mathrm{ddim})} f 2^i$ (by arguments identical to the proof of item **(ii)** above). So by Lemma I.4(i) $B(u, 13 \cdot 2^i)$ intersected edges of $\mathrm{MST}^{NR}(G)$ of weight $2^{O(\mathrm{ddim})} f 2^i$.

We now add some more points to G': Set $j = i - a \log \mathrm{ddim}$ for some absolute constant $a > 2$ to be specified below. Add to G' all j-level points in $B(u, 13 \cdot 2^i)$. Further, choose radius $r \in [12 \cdot 2^i + 72 \cdot 2^j, 13 \cdot 2^i - 72 \cdot 2^j]$ satisfying $w(\mathrm{MST}^{NR}(G) \cap A^*(u, r - 72 \cdot 2^j, r + 72 \cdot 2^j) \le \frac{1}{4} w(\mathrm{MST}^{NR}(G)) \cap B^*(u, r - 72 \cdot 2^j))$ – that is, the weight of $\mathrm{MST}^{NR}(G)$ inside the annulus is a fraction of that inside the hollow. (Such a value r must in fact exist: There are $\Theta(2^i/2^j) = \Theta(2^a \,\mathrm{ddim})$ non-intersecting annuli. If each contained edges of $\mathrm{MST}^{NR}(G)$ weighing a fraction at least $\frac{1}{4}$ of the edge-weight within their hollow, the total edge-weight within $B(u, 13 \cdot 2^i)$ would be $(1 + \frac{1}{4})^{2^a \,\mathrm{ddim}} f 2^i$; for a sufficiently large choice of a, this exceed the upper-bound proved above.) Add to G' all points of the smaller annulus $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$, as well as all points outside outer edge of the annulus – in short, all points outside $B(u, r - 24 \cdot 2^j)$.

To prove the first part of item **(i)**, note that $F(u, i) \cap G'$ contains only points of level j or higher, and these can connected by an MST of weight $(2^i/2^j)^{O(\mathrm{ddim})} 2^i = \mathrm{ddim}^{O(2^a \,\mathrm{ddim})}\, 2^i \ll \frac{f}{10} 2^i$. The final inequality follows when f is chosen to be sufficiently large with respect to a. Then by Lemma I.3, a connecting net-respecting MST has weight less than $\frac{f}{4} 2^i$.

To prove the second part of item **(i)**, we will construct a net-respecting spanning graph for G' with smaller weight than $\mathrm{MST}^{NR}(G)$. Recall that G and G' differ only on points within $B(u, r - 24 \cdot 2^i)$. We add to the spanning graph of G' all edges of $\mathrm{MST}^{NR}(G)$ fully outside of $B(u, r + 24 \cdot 2^i)$ (and this is outside the smaller annulus $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$). We also add to G' those edges with only a single endpoint inside $B(u, r + 24 \cdot 2^i)$, and these net-respecting edges must be incident on the annulus $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$ if they are shorter than $48 \cdot 2^j$, or on j-level points in $(u, r - 24 \cdot 2^j)$ if they are longer. We then claim that the points of G' inside the ball $B(u, r + 24 \cdot 2^j)$ can be covered by a graph whose weight is lighter than that of $\mathrm{MST}^{NR}(G) \cap B^*(u, r + 24 \cdot 2^i)$ by at least $\frac{f}{8} 2^i$, from which the second part of the item follows.

First note that a graph connecting all j-level points in $B(u, 13 \cdot 2^i)$ has weight much less than $\frac{f}{40} 2^i$ for appropriately chosen f (as above in the proof of the first part of **(i)**), and so by Lemma I.3 a net-respecting MST connecting only these points has weight much less than $\frac{f}{16} 2^i$. Further, all points of the annulus $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$ can be connected to some j-level point at total cost at most $w(\mathrm{MST}^{NR}(G) \cap A^*(u, r - 72 \cdot 2^j, r + 72 \cdot 2^j))$ (which was chosen above to be at most $\frac{1}{4} w(\mathrm{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j)))$: The annulus cuts $w(\mathrm{MST}^{NR}(G))$ into disjoint paths. For each such path, if it is incident on a j-level (or higher level) point then we are done. Otherwise, consider the tail of this path, the final segment exiting the smaller annulus $A(u, r - 24 \cdot 2^j, r + 24 \cdot 2^j)$. Since $\mathrm{MST}^{NR}(G)$ is net-respecting, and the path touches only net-points of level less than j, the tail must be of length at least $24 \cdot 2^j$, since it cannot jump immediately to a point outside the larger annulus. Cut off this tail, and instead add to the path a much shorter tail of length at most $2 \cdot 2^j$ connecting to the closest j-level point. So the total cost of the spanning graph for $G' \cap B(u, r + 24 \cdot 2^j)$ is at most $\frac{f}{16} 2^i + \frac{1}{4} w(\mathrm{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j))$, while $w(\mathrm{MST}^{NR}(G) \cap B^*(u, r + 24 \cdot 2^j))$ is of course at least $w(\mathrm{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j))$, a difference of $\frac{3}{4} w(\mathrm{MST}^{NR}(G) \cap B^*(u, r - 72 \cdot 2^j)) - \frac{f}{16} 2^i \ge \frac{3}{4} w(\mathrm{MST}^{NR}(G) \cap B^*(u, 12 \cdot 2^i)) - \frac{f}{16} 2^i \ge \frac{3}{4} \cdot \frac{f}{4} 2^i - \frac{f}{16} 2^i = \frac{f}{8} 2^i$. ■

By using the technique of Lemma IV.1 to repeatedly spin off relatively sparse areas of the graph, we can decompose the entire graph into relatively sparse areas, as in the following theorem:

Theorem IV.2. *Any graph G with aspect ratio Δ and equipped with a hierarchy, may be segmented into a set of subgraphs $\mathcal{D}$ ($\cup_i D_i = G$) with the following properties, for any $c \ge 2$ and for f as above:*

- (i) *Sparsity: For all $D_i \in \mathcal{D}$, $\mathrm{MST}(D_i)$ is $c^{O(\mathrm{ddim})} f$-sparse.*
- (ii) *Weight: $\sum_i w(\mathrm{MST}^{NR}(D_i)) = c^{O(\mathrm{ddim})} w(\mathrm{MST}^{NR}(G))$*
- (iii) *Neighbor Proximity: For every c-neighbor point pair in the hierarchy of G, the pair is found together in at least one set $D_i \in \mathcal{D}$.*
- (iv) *Point occurrence: Each point of G appears in at most $c^{O(\mathrm{ddim})} \log \Delta$ different sets in $\mathcal{D}$.*

The segmentation can be computed in time $c^{O(\mathrm{ddim})} n \log n (\log n + \log \Delta)$.

Proof: The technique of Lemma IV.1 requires knownledge of $\mathrm{MST}^{NR}(G)$ (to compute the annulus), so we must first build this graph. We construct the full hierarchical spanner with parameter c. This can all be done in time $c^{O(\mathrm{ddim})} n \log n$, and the resulting graph has only $c^{O(\mathrm{ddim})} n$ edges [GR08b], [GR08a]. We then retain only those that are net-respecting, and build $\mathrm{MST}^{NR}(G)$ using this set, in total time $c^{O(\mathrm{ddim})} n \log n$. We will need to maintain $\mathrm{MST}^{NR}(G)$ under a long series of point deletions, and this can be done in total time $c^{O(\mathrm{ddim})} n \log^2 n$ (see Theorem 6 in [HdLT01]).

Beginning at the lowest hierarchical level L and iterating upwards, for each i-level net-point u we compute $F(u,i)$ and construct $\mathrm{MST}^{NR}(F(u,i))$, applying the segmentation technique of Lemma IV.1 if necessary. The subset D is added to $\mathcal{D}$, and we iterate the procedure on subset G' until there are no more heavy neighborhoods, at which point the final subset G' is added to $\mathcal{D}$ as well.

The sparsity of sets in $\mathcal{D}$ and neighbor proximity follow immediately from Lemma IV.1(ii) and (iii), respectively. The bound on point occurences follows from the packing property and height of the hierarchy, and a corollary of this bound is that $c^{O(\mathrm{ddim})} n \log n \log \Delta$ time suffices to compute all graphs $\mathrm{MST}^{NR}(F(u,i))$. For the weight, let G_i be the graph in the i-th iteration, and $w_i = w(\mathrm{MST}^{NR}(G_{i+1})) - w(\mathrm{MST}^{NR}(G_i))$. Clearly $\sum_i w_i \leq \mathrm{MST}^{NR}(G)$. By Lemma IV.1(i) and (ii), $\sum_i w(\mathrm{MST}(D_i)) = \sum_i c^{O(\mathrm{ddim})} w_i = c^{O(\mathrm{ddim})} w(\mathrm{MST}^{NR}(G))$. ■

We can now complete the proof of Theorem I.1:

Proof: Fix $c = 64/\varepsilon$. Given graph G, we build a full hierarchy for the entire space, and add to the spanner R the edges of the complete hierarchical $(1+\frac{\varepsilon}{4})$-stretch spanner of Lemma II.1 on all levels below $i = H - \log(n^2)$. Note that $2^H \leq \mathrm{diam}(S)$, and so the longest edge added by this construction is of length $O(1/\varepsilon) \cdot 2^{H-\log(n^2)} = O(\mathrm{diam}(S)/\varepsilon n^2)$. So the entire cost of all these edges is less than $O(1/\varepsilon) \cdot \mathrm{MST}(G)$. We then remove from consideration all net-points below level $i = H - \log(n^2)$, set $L = i$, and focus on the remaining $M = H - L = O(\log n)$ levels. Note that the aspect ratio of this set is only $\Delta = n^{O(1)}$.

We apply Theorem IV.2 to decompose the remaining graph into all sparse sets in time $(\mathrm{ddim}/\varepsilon)^{O(\mathrm{ddim})} n \log^2 n$, and construct the $(1+\frac{\varepsilon}{4})$-stretch spanner of Theorem III.4 for each sparse subset. Since any point of G appears in most $(1/\varepsilon)^{O(\mathrm{ddim})} \log n$ sets (Theorem IV.2(iv)) the total time to construct all these spanners is $(\mathrm{ddim}/\varepsilon)^{O(\mathrm{ddim})} n \log^2 n$. Then the stretch between any pair of c-neighbors is $(1+\frac{\varepsilon}{4})$, and it follows from Lemma II.2 that the total graph stretch is $(1+\varepsilon)$.

The weight guarantee follows from Theorem IV.2(ii) in conjunction with Theorem III.4(i). ■

Open problems.: Our results suggest several avenues for further research. Can W_D be reduced to $\varepsilon^{-O(\mathrm{ddim})}$, to match what is known for Euclidean space? Also, does the simple greedy hierarchical spanner alone yield a light spanner? One can also improve the runtime of our construction.

REFERENCES

[ACC+96] Srinivasa Arikati, Danny Z. Chen, L.Paul Chew, Gautam Das, Michiel Smid, and Christos D. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In Josep Diaz and Maria Serna, editors, *Algorithms . ESA '96*, volume 1136 of *Lecture Notes in Computer Science*, pages 514–528. Springer Berlin Heidelberg, 1996.

[ADD+93] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9:81–100, January 1993.

[ADM+95] Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel Smid. Euclidean spanners: Short, thin, and lanky. In *27th annual ACM symposium on Theory of computing*, pages 489–498, 1995.

[AMS94] S. Arya, D.M. Mount, and M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 703–712, Nov 1994.

[AMS99] S. Arya, D. M. Mount, and M. Smid. Dynamic algorithms for geometric spanners of small diameter: Randomized solutions. *Computational Geometry: Theory and Applications*, 13:91–107, 1999.

[Aro98] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45:753–782, 1998.

[AS97] S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. *Algorithmica*, 17(1):33–54, 1997.

[Ass83] P. Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[BG13] Yair Bartal and Lee-Ad Gottlieb. A linear time approximation scheme for euclidean tsp. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 698–706, 2013.

[BGK12] Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 663–672, 2012. Available at `http://arxiv.org/abs/1112.0699`.

[BGM04] P. Bose, J. Gudmundsson, and P. Morin. Ordered theta graphs. *Computational Geometry: Theory and Applications*, 28:11–18, 2004.

[CG06] Richard Cole and Lee-Ad Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *38th annual ACM symposium on Theory of computing*, pages 574–583, 2006.

[CK95] P. B. Callahan and S.R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. ACM*, 42:67–90, 1995.

[CLNS15] T.-H. Hubert Chan, Mingfei Li, Li Ning, and Shay Solomon. New doubling spanners: Better and simpler. *SIAM Journal on Computing*, 44(1):37–53, 2015.

[DES08] Y. Dinitz, M. Elkin, and S. Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 519–528, Oct 2008.

[DHM10] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010.

[DHN93] Gautam Das, Paul Heffernan, and Giri Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Proceedings of the Ninth Annual Symposium on Computational Geometry*, SCG '93, pages 53–62, New York, NY, USA, 1993. ACM.

[DN97] Gautam Das and Giri Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. *Int. J. Comput. Geometry Appl.*, 7(4):297–315, 1997.

[DNS95] Gautam Das, Giri Narasimhan, and Jeffrey Salowe. A new way to weigh malnourished Euclidean graphs. In *6th annual ACM-SIAM symposium on Discrete algorithms*, pages 215–222. SIAM, 1995.

[ES13] Michael Elkin and Shay Solomon. Optimal euclidean spanners: Really short, thin and lanky. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 645–654, New York, NY, USA, 2013. ACM.

[GGN06] Jie Gao, Leonidas J. Guibas, and An Nguyen. Deformable spanners and applications. *Comput. Geom. Theory Appl.*, 35(1), 2006.

[GH12] Michelangelo Grigni and Hao-Hsiang Hung. Light spanners in bounded pathwidth graphs. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012*, volume 7464 of *Lecture Notes in Computer Science*, pages 467–477. Springer Berlin Heidelberg, 2012.

[GKK13] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient regression in metric spaces via approximate lipschitz extension. In *Similarity-Based Pattern Recognition - Second International Workshop, SIMBAD 2013, York, UK, July 3-5, 2013. Proceedings*, pages 43–58, 2013.

[GLN02] Joachim Gudmundsson, Christos Levcopoulos, and Giri Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM Journal on Computing*, 31:2002, 2002.

[GR08a] Lee-Ad Gottlieb and Liam Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 591–600, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[GR08b] Lee-Ad Gottlieb and Liam Roditty. An optimal dynamic spanner for doubling metric spaces. In *Proceedings of the 16th annual European symposium on Algorithms*, ESA '08, pages 478–489. Springer-Verlag, 2008.

[GS14] Lee-Ad Gottlieb and Shay Solomon. Light spanners for snowflake metrics. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, New York, NY, USA, 2014. ACM.

[HdLT01] Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48(4):723–760, 2001.

[HM06] Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.

[KL04] Robert Krauthgamer and James R. Lee. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 791–801, January 2004.

[KPX08] Iyad A. Kanj, Ljubomir Perkovi, and Ge Xia. Computing lightweight spanners locally. In *Distributed Computing*, volume 5218 of *Lecture Notes in Computer Science*, pages 365–378. Springer Berlin Heidelberg, 2008.

[Pan13] D. Panchenko. Theory of probability- empirical process and kolmogorov's chaining. Manuscript, 2013.

[RS98] Satish B. Rao and Warren D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *30th annual ACM symposium on Theory of computing*, pages 540–550. ACM, 1998.

[Sal91] J. S. Salowe. Constructing multidimensional spanner graphs. *Int. J. Comput. Geometry Appl*, 1(2):99–107, 1991.

[Smi09] Michiel Smid. The weak gap property in metric spaces of bounded doubling dimension. In *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of his 60th Birthday*, pages 275–289. Springer-Verlag, 2009.

[Smi10] Michiel Smid. On some combinatorial problems in metric spaces of bounded doubling dimension. Manuscript, available at `http://people.scs.carleton.ca/ michiel/research.html`, 2010.

[Soa94] J. Soares. Approximating Euclidean distances by small degree graphs. *Discrete & Computational Geometry*, 11:213–233, 1994.

[Sol11] Shay Solomon. An optimal-time construction of sparse euclidean spanners with tiny diameter. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 820–839. SIAM, 2011.

[Sol14] Shay 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*, STOC '14, pages 363–372, New York, NY, USA, 2014. ACM.

[Tal04] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *36th annual ACM symposium on Theory of computing*, pages 281–290. ACM, 2004.

[Vai91] P. M. Vaidya. A sparse graph almost as good as the complete graph on points in K dimensions. *Discrete & Computational Geometry*, 6:369–381, 1991.