Distributed Construction of Lightweight Spanners for Unit Ball Graphs

David Eppstein
Department of Computer Science, University of California, Irvine, CA, USA

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

Abstract

Resolving an open question from 2006 [14], we prove the existence of light-weight bounded-degree spanners for unit ball graphs in the metrics of bounded doubling dimension, and we design a simple $\mathcal{O}(\log^* n)$-round distributed algorithm in the LOCAL model of computation, that given a unit ball graph G with n vertices and a positive constant $\epsilon < 1$ finds a $(1+\epsilon)$-spanner with constant bounds on its maximum degree and its lightness using only 2-hop neighborhood information. This immediately improves the best prior lightness bound, the algorithm of Damian, Pandit, and Pemmaraju [13], which runs in $\mathcal{O}(\log^* n)$ rounds in the LOCAL model, but has a $\mathcal{O}(\log \Delta)$ bound on its lightness, where Δ is the ratio of the length of the longest edge to the length of the shortest edge in the unit ball graph. Next, we adjust our algorithm to work in the CONGEST model, without changing its round complexity, hence proposing the first spanner construction for unit ball graphs in the CONGEST model of computation. We further study the problem in the two dimensional Euclidean plane and we provide a construction with similar properties that has a constant average number of edge intersections per node. Lastly, we provide experimental results that confirm our theoretical bounds, and show an efficient performance from our distributed algorithm compared to the best known centralized construction.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Sparsification and spanners; Networks → Network control algorithms

Keywords and phrases spanners, doubling metrics, distributed, topology control

Digital Object Identifier 10.4230/LIPIcs.DISC.2022.21

Related Version *Full Version*: `https://arxiv.org/abs/2106.15234` [24]

1 Introduction

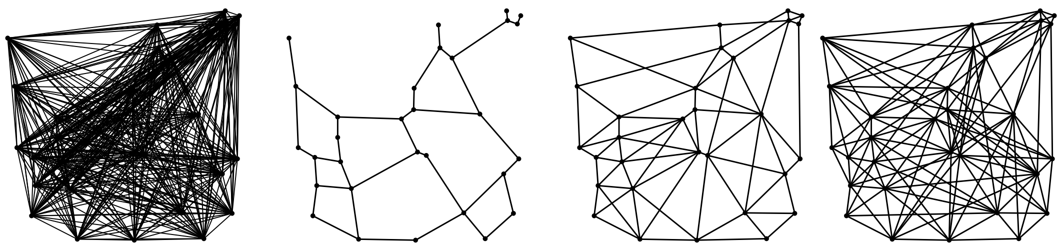
Given a collection of points V in a metric space with doubling dimension d, the weighted *unit ball graph (UBG)* on V is defined as a weighted graph $G(V, E)$ where two points $u, v \in V$ are connected if and only if their metric distance $\|uv\| \leq 1$. The weight of the edge uv of the UBG is $\|uv\|$ if the edge exists. Unit ball graphs in the Euclidean plane are called unit disk graphs (UDGs) and are frequently used to model ad-hoc wireless communication networks, where every node in the network has an effective communication range R, and two nodes are able to communicate if they are within a distance R of each other.

Spanners are sub-graphs of the input graph whose pair-wise distances approximate distances in the input graphs, while having fewer edges than complete graphs. Given a weighted graph G, a t-spanner on G can be defined as a graph S that has $V(G)$ as its set of vertices, while $E(S) \subseteq E(G)$ and the following inequality is satisfied for every pair of vertices $u, v \in V(G)$:

$$\mathrm{dist}_S(u, v) \leq t \cdot \mathrm{dist}_G(u, v)$$

where $\mathrm{dist}_S(u, v)$ (or $\mathrm{dist}_G(u, v)$) is the length of the shortest path between u and v using the edges in S (or G, respectively). We call this inequality the *bounded stretch property*. Because of this inequality, t-spanners provide a t-approximation for the pairwise distances between the vertices in G. The parameter $t > 1$ is called the *stretch factor* or *spanning ratio* of the spanner and determines how accurate the approximate distances are; spanners having smaller stretch factors are more accurate.

Spanners can be specifically defined on any graph coming from a metric space, where a heavy or undesirable network is given and finding a sparse and light-weight spanner and working with it instead of the actual network makes the computation easier and faster (Figure 1). In particular, lightweight spanners have been gained extreme attention in the geometric setting and in the metrics with bounded doubling dimension [29, 7, 6], which is a generalization of the former. The problem of finding sparse light-weight spanners in these spaces has appeared in many areas of computer science, including communication network design and distributed computing. These subgraphs have few edges and are easy to construct, leading them to appear in a wide range of applications since they were introduced [11, 34, 43]. In wireless ad hoc networks t-spanners are used to design sparse networks with guaranteed connectivity and guaranteed bounds on routing length [3]. In distributed computing spanners provide communication-efficiency and time-efficiency through the sparsity and the bounded stretch property [5, 19, 4, 20]. There has also been extensive use of geometric spanners in the analysis of road networks [21, 1, 10]. In robotics, geometric spanners helped motion planners to design near-optimal plans on a sparse and light subgraph of the actual network [16, 40, 15]. Spanners have many other applications including computing almost shortest paths [17, 12, 44, 26], and overlay networks [8, 47, 32].



(a) Complete graph. **(b)** 2-spanner. **(c)** 1.2-spanner. **(d)** 1.05-spanner.

Figure 1 A comparison of the complete graph on 30 random points on the plane with spanners of stretch 2, 1.2, and 1.05 on the same point set.

The special case where the underlying graph is a unit ball graph is motivated by the application of unit ball graphs in modeling wireless and ad-hoc networks, where the communication of the nodes are limited by their physical distances. The problem of finding sparse lightweight spanners for unit ball graphs in this settings translates into efficient topology control algorithms. Thus the necessity of a connected and energy-efficient topology for high-level routing protocols led researchers to develop many spanning algorithms for ad-hoc networks and in particular, UDGs. But the decentralized nature of ad-hoc networks demands that these algorithms be local instead of centralized. In these applications, it is important that the resulting topology is connected, has a low weight, and has a bounded degree, implying also that the number of edges is linear in the number of vertices.

Several known *proximity graphs* have been studied for this purpose, including the relative neighborhood graph (RNG), Gabriel graph (GG), Delaunay graph (DG), and Yao graph (YG). It is well-known that these proximity graphs are sparse and they can be calculated locally, using only the information from a node's neighborhood. But further analysis shows that they have poor bounds on at least one of the important criteria: maximum vertex degree, total weight, and stretch-factor [39].

Researchers have modified these constructions to fulfill the requirements. Li, Wan, and Wang [39] introduced a modified version of the Yao graph to resolve the issue of unbounded in-degree while preserving a small stretch-factor, but they left as an open question whether there exists a construction whose total weight is also bounded by a constant factor of the weight of the minimum spanning tree. The localized Delaunay triangulation (LDT) [38] and local minimum spanning tree (LMST) [37] were two other efforts in this way which failed to bound the total weight of the spanner. Hence bounding the weight became the main challenge in designing efficient spanners. The commonly used measure for the weight of the spanners is *lightness*, which is defined as the weight of the spanner divided by the weight of the minimum spanning tree.

In the distributed setting in particular, Gao, Guibas, Hershberger, Zhang, and Zhu [28] introduced restricted Delaunay graph (RDG), a planar distributed spanner construction for unit disk graphs in the two dimensional Euclidean plane that possessed a constant stretch-factor, leaving the weight of the spanner unstudied. Later Kanj, Perković, and Xia [33] presented the first local spanner construction for unit disk graphs in the two dimensional Euclidean plane, which also was planar and had constant bounds on its stretch-factor, maximum degree, and lightness. Their construction was also based on the Delaunay triangulation of the point set and required information from k-th hop neighbors of every node, for some constant k that depended on the input parameters.

In 2006, Damian, Pandit, and Pemmaraju [14] designed a distributed construction for $(1+\epsilon)$-spanners of the UBGs lying in *d-dimensional Euclidean space*. Their algorithm ran in $\mathcal{O}(\log^* n)$ rounds of communication and produced a $(1+\epsilon)$-spanner with constant bounds on its maximum degree and lightness. They used the so-called *leapfrog property* to prove the constant bound on the lightness of the spanner, which does not hold for the spaces of bounded doubling dimension in general. Instead, they showed in another work [13] that the weight of their spanner in the spaces of bounded doubling dimension is bounded by a factor $\mathcal{O}(\log \Delta)$ of the weight of the minimum spanning tree, where Δ is the ratio of the length of the longest edge in the unit ball graph divided by the length of its shortest edge. Besides these, their algorithm requires the knowledge of $\mathcal{O}(\frac{1}{\alpha-1})$-hop neighborhood of the nodes, which is costly in the CONGEST model of distributed computing, the more accepted and practical model than the LOCAL model of computation.

In the 3D Euclidean space, Jenkins, Kanj, Xia, and Zhang [31] designed the first localized bounded-degree $(1+\epsilon)$-spanner for unit ball graphs. They also presented a lightweight construction which possessed constant bounds on its stretch-factor and maximum degree. These algorithms again required k-th hop neighborhood information for every node, for a constant k that depended on the input parameters. Although these constructions were local, i.e. they ran in constant rounds of communication, they relied heavily on Euclidean transformations which made them inapplicable for other metric spaces.

Finally, Elkin, Filtser, and Neiman [18] studied the topic of lightweight spanners for general graphs and doubling graphs in the CONGEST model of distribution. For general graphs, they presented $(2k-1)\cdot(1+\epsilon)$-spanners with lightness $\mathcal{O}(k \cdot n^{1/k})$ in $\tilde{\mathcal{O}}(n^{0.5+1/(4k+2)}+D)$ rounds, where n is the number of vertices and D is the hop-diameter of the graph. For doubling graphs, they presented a $(1+\epsilon)$-spanner with lightness $\epsilon^{-\mathcal{O}(1)} \log n$ in $(\sqrt{n}+D)\cdot n^{o(1)}$ rounds of communication. Although these constructions are more general than the constructions of [13] and they perform in a more restricted model (CONGEST), they do not imply a superior result in the specific case of unit ball graphs in doubling metrics.

Apart from being a generalization of the Euclidean space, the importance of the spaces of bounded doubling dimension comes from the fact that a small perturbation in the pairwise distances does not affect the doubling dimension of the point set by much, while it can change their Euclidean dimension significantly, or the resulting distances might not even be embeddable in Euclidean metrics at all [9]. This makes these metrics of bounded doubling dimension to be more applicable in real-world scenarios. On the other hand, geometric arguments are considered as a strong tool for proofs of sparsity and lightness bounds in Euclidean spaces, but in doubling spaces the only available tool besides metric properties, is the packing argument which is directly followed from the definition of the doubling dimension. Therefore, the sparsity and lightness results are more difficult to achieve in the spaces of bounded doubling dimension.

Since the work of Damian, Pandit, and Pemmaraju [13] in 2006, it has remained open whether UBGs in the spaces of bounded doubling dimension possess lightweight bounded-degree $(1+\epsilon)$-spanners and whether they can be found efficiently in a distributed model of computation. On the other hand, the construction of [13] requires complete information about the nodes in $\mathcal{O}(\frac{1}{\alpha-1})$ hops away, for some constant α. Acquiring this information is costly in the CONGEST model of computation, which is a more accepted model in distributed computing. Therefore, another open question arising from this line of work is to study the round complexity of the aforementioned problem in the CONGEST model. In this paper, we resolve both of these long-standing open questions by presenting centralized and distributed algorithms, both in the LOCAL, and the CONGEST model, for the purpose of finding such spanners.

1.1 Contributions

We have two main contributions in this paper. First, we resolve the proposed open question that has remained open for more than a decade, and we prove the existence of light-weight bounded-degree $(1+\epsilon)$-spanners of unit ball graphs in the spaces of bounded doubling dimension. Our construction has constant bounds on its maximum degree and its lightness, and it can be built in $\mathcal{O}(\log^* n)$ rounds of communication in the LOCAL model of computation, where n is the number of vertices.

Second, we propose the first lightweight spanner construction for unit ball graphs in the CONGEST model of computation. Even if we restrict our scope to the two dimensional Euclidean plane, where we see most of the applications of unit disk graphs, prior to this work there was known CONGEST algorithm for finding light spanners of unit disk graphs. We achieve this construction by making adjustments on our construction for the LOCAL model to make it work in the CONGEST model in the same asymptotic number of rounds. The bounds on the lightness and maximum degree of our spanner remain the same in this model.

Besides these main results, we modify these constructions for the two dimensional Euclidean plane in order to have a linear number of edge intersections in total, implying a constant average number of edge intersections per node. This is motivated by the observation that a higher intersection per edge causes a higher chance of interference between the corresponding endpoints. To the best of our knowledge, this is the first distributed low-stretch low-intersection spanner construction for unit disk graphs.

A more detailed version of our results can be found in the following theorems. First, we introduce a centralized algorithm Centralized-Spanner that,

▶ **Theorem 8.** *Given a weighted unit ball graph G in a metric of bounded doubling dimension and a constant $\epsilon > 0$, the spanner returned by Centralized-Spanner(G,ϵ) is a $(1+\epsilon)$-spanner of G and has constant bounds on its lightness and maximum degree. These constant bounds only depend on ϵ and the doubling dimension.*

We use this centralized construction to propose the distributed construction DISTRIBUTED-SPANNER in the LOCAL model of computation,

▶ **Theorem 16.** *Given a weighted unit ball graph G with n vertices in a metric of bounded doubling dimension and a constant $\epsilon > 0$, the algorithm DISTRIBUTED-SPANNER(G,ϵ) runs in $\mathcal{O}(\log^* n)$ rounds of communication in the LOCAL model of computation, and returns a $(1+\epsilon)$-spanner of G that has constant bounds on its lightness and maximum degree. These constant bounds only depend on ϵ and the doubling dimension.*

Next, we study the problem in the CONGEST model of computation. Our distributed construction DISTRIBUTED-SPANNER requires complete information about 2-hop neighborhood of a selected set of vertices, which is not easy to acquire in the CONGEST model. The same issues exists in the distributed algorithm of [13], where they aggregate information about the nodes that are $\mathcal{O}(\frac{1}{\alpha-1})$ hops away, for some constant α. A simple approach for aggregating 2-hop neighborhoods would require $\mathcal{O}(d)$ rounds of communication in the CONGEST model, which can be as large as $\Omega(n)$ if the input graph is dense. In our next theorem, we break this barrier by making some adjustments for our algorithm to work in the CONGEST model of computation. Despite adding to the complexity of the algorithm itself, we prove that the round complexity of our new algorithm, CONGEST-SPANNER, would still be bounded by $\mathcal{O}(\log^* n)$.

▶ **Theorem 21.** *Given a weighted unit ball graph G with n vertices in a metric of bounded doubling dimension and a constant $\epsilon > 0$, the algorithm CONGEST-SPANNER(G,ϵ) runs in $\mathcal{O}(\log^* n)$ rounds of communication in the CONGEST model of computation, and returns a $(1+\epsilon)$-spanner of G that has constant bounds on its lightness and maximum degree. These constant bounds only depend on ϵ and the doubling dimension.*

The rest of the contributions are included in the full version of this paper due to page limits. In the full version, We study the problem in the case of the two dimensional Euclidean plane, where the greedy spanner on a complete weighted graph is known to have constant upper bounds on its lightness [27], maximum degree, and average number of edge intersections per node [23]. We observe that a simple change on the this algorithm can extend these results for unit disk graphs as well. We call this modified algorithm CENTRALIZED-EUCLIDEAN-SPANNER and we show that

▶ **Theorem 22.** *Given a weighted unit disk graph G in the two dimensional Euclidean plane and a constant $\epsilon > 0$, the spanner returned by CENTRALIZED-EUCLIDEAN-SPANNER(G,ϵ) is a $(1+\epsilon)$-spanner of G and has constant bounds on its lightness, maximum degree, and the average number of edge intersections per node. These constant bounds only depend on ϵ and the doubling dimension.*

We use the aforementioned construction to propose DISTRIBUTED-EUCLIDEAN-SPANNER, a specific distributed low-intersection construction for the case of the two dimensional Euclidean plane that preserves the previously mentioned properties and adds the low-intersection property.

▶ **Theorem 26.** *Given a weighted unit disk graph G with n vertices in the two dimensional Euclidean plane and a constant $\epsilon > 0$, the algorithm DISTRIBUTED-EUCLIDEAN(G,ϵ) runs in $\mathcal{O}(\log^* n)$ rounds of communication and returns a bounded-degree $(1+\epsilon)$-spanner of G that has constant bounds on its lightness, maximum degree, and the average number of edge intersections per node. These constant bounds only depend on ϵ and the doubling dimension.*

We also prove that the last construction possesses sublinear separators and a separator hierarchy in the two dimensional Euclidean plane. We generalize this result to work for higher dimensions of Euclidean spaces. Finally, we provide experimental results on random point sets in the two dimensional Euclidean plane that confirm the efficiency of our distributed construction.

2 Preliminaries

2.1 Doubling metrics

We start by recalling the definition of the doubling dimension of a metric space,

▶ **Definition 1** (doubling dimension). *The doubling dimension of a metric space is the smallest d such that for any $R > 0$, any ball of radius R can be covered by at most 2^d balls of radius $R/2$.*

We say a metric space has *bounded doubling dimension* if its doubling dimension is upper bounded by a constant. Besides the triangle inequality, which is intrinsic to metric spaces, the *packing lemma* is an essential tool for the metrics of bounded doubling dimension. This lemma states that it is impossible to pack more than a certain number of points in a ball of radius $R > 0$ without making a pair of points' distance less than some $r > 0$.

▶ **Lemma 2** (Packing Property). *In a metric space of bounded doubling dimension d, let X be a set of points with minimum distance r, contained in a ball of radius R. Then $|X| \leq \left(\frac{4R}{r}\right)^d$.*

Proof. This is a well-known fact, see e.g. [46]. ◀

2.2 Spanners for complete graphs

For a weighted graph G in a metric space, where every edge weight is equal to the metric distance of its endpoints, a t-spanner is defined in the following way,

▶ **Definition 3** (t-spanner). *A t-spanner of a weighted graph G is a subgraph S of G that for every pair of vertices x, y in G,*

$$\text{dist}_S(x, y) \leq t \cdot \text{dist}_G(x, y)$$

where $\text{dist}_A(x, y)$ is the length of a shortest path between x and y in A. The lightness of S is defined as $\mathbf{w}(S)/\mathbf{w}(MST)$ where $\mathbf{w}$ is the weight function and MST is the minimum spanning tree in G.

In other words, a t-spanner approximates the pairwise distances within a factor of t. Spanners were studied for complete weighted graphs first, and several constructions were proposed to optimize them with respect to the number of edges and total weight. Among these constructions, *greedy spanners* [2] are known to out-perform the others.

A greedy spanner (Figure 1) can be constructed by running the greedy spanner algorithm (Algorithm 1) on a set of points V in a metric space. This short procedure adds edges one at a time to the spanner it constructs, in ascending order by length. For each pair of vertices, in this order, it checks whether the pair already satisfies the distance inequality using the edges already added. If not, it adds a new edge connecting the pair. Therefore, by construction, each pair of vertices satisfies the inequality, either through previous edges or (if not) through the newly added edge. The resulting graph is therefore a t-spanner.

Algorithm 1 The naive greedy spanner algorithm.

procedure NAIVE-GREEDY(V)
  Let S be a graph with vertices V and edges $E = \{\}$
  for each pair $(P, Q) \in V^2$ in increasing order of $\|PQ\|$ **do**
    if $\text{dist}_S(P, Q) > t \cdot \text{dist}(P, Q)$ **then**
      Add edge PQ to E
  return S

Despite the simplicity of Algorithm 1, Farshi and Gudmundsson [25] observed that in practice, greedy spanners are surprisingly good in terms of the number of edges, weight, maximum vertex degree, and also the number of edge crossings in the two dimensional Euclidean plane. All of these properties have been proven rigorously so far. Filster and Solomon [27] proved that greedy spanners have size and lightness that is optimal to within a constant factor for worst-case instances. They also achieved a near-optimality result for greedy spanners in spaces of bounded doubling dimension. Borradaile, Le, and Wulff-Nilsen [7] recently proved optimality for doubling metrics, generalizing a result of Narasimhan and Smid [41], and resolving an open question posed by Gottlieb [29], and Le and Solomon showed that no geometric t-spanner can do asymptotically better than the greedy spanner in terms of number of edges and lightness [36].

In a recent work, Eppstein and Khodabandeh [23] showed that the number of edge crossings of the greedy spanner in the two dimensional Euclidean plane is linear in the number of vertices. Moreover, they proved that the crossing graph of the greedy spanner has bounded degeneracy, implying the existence of sub-linear separators for these graphs [22]. This, together with the well-known fact that greedy spanners have bounded-degree in the two dimensional Euclidean plane, makes greedy spanners more practical in this particular metric space.

Although the degree of the greedy spanner is bounded in the two dimensional Euclidean plane, it is known that there exist n-point metric spaces with doubling dimension 1 where the greedy spanner has maximum degree $n - 1$ [27]. Gudmundsson, Levcopoulos, and Narasimhan [30] devised a faster algorithm that was later proven to have bounded degree as well as constant lightness and linear number of edges [27]. We call this algorithm APPROXIMATE-GREEDY in this paper, and we make use of it in our algorithms for the metrics of bounded doubling dimension, while we take advantage of the extra low-intersection property of NAIVE-GREEDY in the two dimensional Euclidean plane.

2.3 Unit ball graphs

We formally define a unit ball graph on a set of points V in the following way,

▶ **Definition 4** (unit ball graph). *Given a set of points V in a metric space, the unit ball graph G on V contains V as its vertex set and every two vertices $x, y \in V$ are connected if and only if $\|xy\| \leq 1$. The weight of an edge (x, y) is equal to $\|xy\|$ if the edge exists.*

Unit ball graphs are an important subclass of the graphs called *growth-bounded graphs*, which only limit the number of independent nodes in every neighborhood, a property that holds for UBGs due to the packing property.

Kuhn, Moscibroda, and Wattenhofer [35] presented a $\mathcal{O}(\log^* n)$-round distributed algorithm for finding a maximal independent set (MIS) of a unit ball graph graph in a space with bounded doubling dimension. This result was later generalized by Schneider and Wattenhofer [45] for growth-bounded graphs. Throughout the paper we refer to their algorithm by

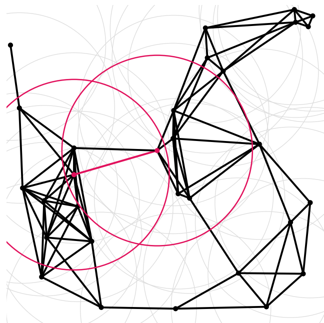


Figure 2 The unit disk graph on the same point set introduced earlier. The red disks intersect, therefore there is an edge between their centers.

Maximal-Independent. It turns out that Maximal-Independent will be a key ingredient of our distributed algorithms, as well as their bottleneck in terms of the number of rounds. This means that if a maximal independent set is known beforehand, our algorithms can be executed fully locally, in constant number of rounds.

In section 3 we prove the existence of $(1+\epsilon)$-spanners with constant bounds on the maximum degree and the lightness by introducing an algorithm that finds such spanners in a centralized manner. In section 4 we propose a distributed construction that delivers the same features through a $\mathcal{O}(\log^* n)$-round algorithm. In the full version, we consider the special case of two dimensional Euclidean plane and we design centralized and distributed algorithms to construct a spanner that has the extra low-intersection property, making it more suitable for practical purposes.

3 Centralized Construction

In this section we propose a centralized construction for a light-weight bounded-degree $(1+\epsilon)$-spanner for unit ball graphs in a metric of bounded doubling dimension. Later in section 4 we use this centralized construction to design a distributed algorithm that delivers the same features.

It is worth mentioning that the greedy spanner would be a $(1+\epsilon)$-spanner of the UBG if the algorithm stops after visiting the pairs of distance at most 1, and it even has a lightness bounded by a constant, but as we mentioned earlier, there are metrics with doubling dimension 1 in which its degree may be unbounded.

To construct a lightweight bounded-degree $(1+\epsilon)$-spanner of the unit ball graph, we start with the spanner of [30], called Approximate-Greedy, which is returns a spanner of the complete graph. It is proven in [41] that Approximate-Greedy has the desired properties, i.e. bounded-degree and lightness, for complete weighted graphs in Euclidean metrics, but as stated in [27], the proof only relies on the triangle inequality and packing argument which both work for doubling metrics as well. Therefore, we may safely assume that Approximate-Greedy finds a light-weight bounded-degree $(1+\epsilon)$-spanner of the complete weighted graph defined on the point set. The main issue is that the edges of length more than 1 are not allowed in a spanner of the unit ball graph on the same point set. Therefore, a replacement procedure is needed to substitute these edge with edges of length at most 1. Peleg and Roditty [42] introduced a refinement process which removes the edges of length larger than 1 from the spanner and replaces them with three smaller edges to make the output a subgraph of the UBG. The main issue with their approach is that it can lead to vertices having unbounded degrees in the spanner, therefore missing an important feature. Here, we introduce our own refinement process that not only replaces edges of larger than 1 with smaller edges and makes the spanner a subgraph of the unit ball graph, but also guarantees a constant bounded on the degrees of the resulting spanner.

3.1 The algorithm

In the first step of the algorithm (Algorithm 2) we choose $\epsilon' = \epsilon/36$, a smaller stretch parameter than ϵ, to cover the errors that future steps might inflict to the spanner. Then we call the procedure Approximate-Greedy on the set of vertices V to calculate a light-weight bounded-degree $(1+\epsilon')$-spanner S of the complete weighted graph on V. This spanner might contain edges of length larger than 1, which we will replace by some edges of length at most 1 in the future steps.

Since an edge of length larger than $1+\epsilon'$ in S cannot participate in the shortest path between any two adjacent vertices in G, we simply remove and discard them from the spanner. Then for every remaining edge $e = (u, v)$ of length in the range $(1, 1+\epsilon']$ we find an edge (x, y) of the original graph G so that $\|ux\| \leq \epsilon'$ and $\|vy\| \leq \epsilon'$. We then replace such an edge e by the edge (x, y). We call the pair (x, y) the *replacement edge* or the *replacement pair* for the edge e. Since this procedure can end up assigning too many replacement edges to a single vertex (x or y in this case) and hence increasing its degree significantly, we perform a simple check before adding a replacement edge; we store the set R of previously added replacement pairs in the memory and if a *weak* replacement pair $(x', y') \in R$ exists, then we prefer it over a newly found replacement pair $(x, y) \notin R$. By weak replacement pair we mean a pair $(x', y') \in R$ that $\|ux'\| \leq 2\epsilon'$ and $\|vy'\| \leq 2\epsilon'$, which is weaker than the definition of the replacement pair. As we later see this weaker notion of replacement pair will help us to bound the degree of the vertices.

After removing the edges of length larger than 1 and replacing the ones in the range $(1, 1+\epsilon']$, we return the spanner to the output.

Algorithm 2 A centralized spanner construction.

Input. A unit ball graph $G(V, E)$ in a metric with doubling dimension d.
Output. A light-weight bounded-degree $(1+\epsilon)$-spanner of G.

procedure Centralized-Spanner(G, ϵ)
    $\epsilon' \leftarrow \epsilon/36$
    $S \leftarrow$ Approximate-Greedy(V, ϵ')
    R $\leftarrow \varnothing$
    for $e = (u, v)$ in S **do**
        if $|e| > 1$ **then**
            Remove e from S
        if $|e| \in (1, 1+\epsilon']$ **then**
            if $\exists(x, y) \in E$ that $\|ux\| \leq \epsilon'$ and $\|vy\| \leq \epsilon'$ **then**
                if $\nexists(x', y') \in R$ that $\|ux'\| \leq 2\epsilon'$ and $\|vy'\| \leq 2\epsilon'$ **then**
                    $S \leftarrow S \cup \{(x, y)\}$
                    $R \leftarrow R \cup \{(x, y)\}$
    return S

3.2 The analysis

Now we prove that the output S of the algorithm is a light-weight bounded-degree $(1+\epsilon)$-spanner of the unit ball graph G. Clearly, after the refinement is done the spanner S is a subgraph of G, so we need to analyze the lightness, the stretch factor, and the maximum degree of the spanner.

First we prove that the stretch-factor of the spanner is indeed bounded by $1+\epsilon$.

▶ **Lemma 5.** *The spanner returned by* Centralized-Spanner *has a stretch factor of* $1+\epsilon$.

The proof of this lemma is moved to Appendix A. Now we analyze the weight of the spanner, proving its constant lightness.

▶ **Lemma 6.** *The spanner returned by* Centralized-Spanner *has a weight of* $\mathcal{O}(1)\mathbf{w}(MST)$.

The proof of this lemma is also in Appendix A. In the final step, we bound the maximum degree of the spanner.

▶ **Lemma 7.** *The spanner returned by* Centralized-Spanner *has bounded degree.*

The proof of this lemma is moved to Appendix A. Putting these together, we can prove Theorem 8.

▶ **Theorem 8** (Centralized Spanner)**.** *Given a weighted unit ball graph G in a metric of bounded doubling dimension and a constant $\epsilon > 0$, the spanner returned by* Centralized-Spanner*(G,ϵ) is a $(1+\epsilon)$-spanner of G and has constant bounds on its lightness and maximum degree. These constant bounds only depend on ϵ and the doubling dimension.*

Proof. Follows directly from Lemma 5, Lemma 6, and Lemma 7. ◀

4 Distributed Construction

In this section we propose our distributed construction for finding a $(1+\epsilon)$-spanner of a unit ball graph using only 2-hop neighborhood information. The spanner returned by our algorithm has constant bounds on its maximum degree and its lightness. This is the first light-weight distributed construction for unit ball graphs in doubling metrics, to the best of our knowledge.

In our distributed construction, we run our centralized algorithm on the 2-hop neighborhoods of a an independent set of the unit ball graph, and we prove that putting these local spanners together will achieve a spanner that possesses the desired properties.

4.1 The algorithm

For the distributed construction we propose Algorithm 3. There is a preprocessing step of finding a maximal independent set I of G, which can be done using the distributed algorithm of [35] in $\mathcal{O}(\log^* n)$ rounds. We refer to this algorithm by Maximal-Independent. Then the Local-Greedy subroutine is run on every vertex $w \in I$ to find a $(1+\epsilon)$-spanner S_w of the 2-hop neighborhood of w, denoted by $\mathcal{N}^2(w)$. At the final step, every $w \in I$ sends its local spanner edges to the corresponding endpoints of every edge. Symmetrically, every vertex listens for the edges sent by the vertices in I and once a message is received, it stores the edges in its local storage. In other words, the final spanner is the union of all these local spanners. We use the centralized algorithm of section 3 for every local neighborhood $\mathcal{N}^2(w)$ to guarantee the bounds that we need.

Algorithm 3 The localized greedy algorithm.

Input. A unit ball graph $G(V, E)$ in a metric with doubling dimension d and an $\epsilon > 0$.
Output. A light-weight bounded-degree $(1+\epsilon)$-spanner of G.

procedure DISTRIBUTED-SPANNER(G, ϵ)
    Find a maximal independent set I of G using [35]
    Run LOCAL-GREEDY on the vertices of G
function LOCAL-GREEDY(vertex w)
    Retrieve $\mathcal{N}^2(w)$, the 2-hop neighborhood information of w
    if $w \in I$ **then**
        $\mathcal{S}_w \leftarrow$ CENTRALIZED-SPANNER($\mathcal{N}^2(w)$, ϵ)
        for $e = (u, v)$ in $\mathcal{S}_w$ **do**
            Send e to u and v
    Listen to incoming edges and store them

Similar to the aforementioned greedy algorithm (Algorithm 1), our algorithm seems very simple in the first sight. But as we see later in this section, proving its properties, particularly its lightness, is a non-trivial task.

4.2 The analysis

Now we show that the spanner introduced in Algorithm 3 possesses the desired properties. First, we show the round complexity of $\mathcal{O}(\log^* n)$.

▶ **Lemma 9.** *DISTRIBUTED-SPANNER can be done in $\mathcal{O}(\log^* n)$ rounds of communication.*

The proof of this lemma is in Appendix B. Next we bound the stretch-factor of the spanner.

▶ **Lemma 10.** *The spanner returned by DISTRIBUTED-SPANNER has a stretch factor of $1+\epsilon$.*

The proof of this lemma is also included in Appendix B. Now we bound the maximum degree of the spanner.

▶ **Lemma 11.** *The spanner returned by DISTRIBUTED-SPANNER has a bounded degree.*

The proof of this lemma is also in Appendix B. In order to bound the lightness of the output, we assume that $\epsilon \leq 1$ and we make a few comparisons. First, for any $w \in I$ we compare the weight of $\mathcal{S}_w$ to the weight of the minimum spanning tree on $\mathcal{N}^2(w)$. Then we compare the weight of the minimum spanning tree on $\mathcal{N}^2(w)$ to the weight of the minimum Steiner tree on $\mathcal{N}^3(w)$, where the required vertices are $\mathcal{N}^2(w)$ and 3-hop vertices are optional. Finally, we compare the weight of this minimum Steiner tree to the weight of the induced subgraph of CENTRALIZED-SPANNER(G, ϵ) on the subset of vertices $\mathcal{N}^3(w)$, which later implies that the overall weight of $\mathcal{S}_w$s is bounded by a constant factor of the weight of the minimum spanning tree on G.

Our first claim is that the weight of $\mathcal{S}_w$ is bounded by a constant factor of the weight of the MST on $\mathcal{N}^2(w)$.

▶ **Corollary 12.** $\mathbf{w}(\mathcal{S}_w) = \mathcal{O}(1)\mathbf{w}(MST(\mathcal{N}^2(w)))$

Proof. Follows from the properties of the centralized algorithm in section 3. ◀

Next we compare $\mathbf{w}(MST(\mathcal{N}^2(w)))$ to the weight of the minimum Steiner tree of $\mathcal{N}^3(w)$ on the required vertices $\mathcal{N}^2(w)$.

► **Lemma 13.** *Define $\mathcal{T}$ to be the optimal Steiner tree on the set of vertices $\mathcal{N}^3(w)$, where only vertices in $\mathcal{N}^2(w)$ are required and the rest of them are optional. Then*

$$\mathbf{w}(MST(\mathcal{N}^2(w))) \leq 2\mathbf{w}(\mathcal{T})$$

The proof of this lemma is included in Appendix B. We then compare the weight of $\mathcal{T}$ to the weight of induced subgraph of CENTRALIZED-SPANNER(G, ϵ) on the subset of vertices $\mathcal{N}^3(w)$. The main observation here is that when $\epsilon \leq 1$ the induced subgraph of the centralized spanner on $\mathcal{N}^3(w)$ would be a feasible solution to the minimum Steiner tree problem on $\mathcal{N}^3(w)$, with the required vertices being the vertices in $\mathcal{N}^2(w)$. This will imply that the weight of the induced subgraph is at least equal to the weight of the minimum Steiner tree.

► **Lemma 14.** *Let $\mathcal{S}^*$ be the output of CENTRALIZED-SPANNER(G, ϵ) and let $\mathcal{S}^*_w$ be the induced subgraph of $\mathcal{S}^*$ on $\mathcal{N}^3(w)$. Then*

$$\mathbf{w}(\mathcal{T}) \leq \mathbf{w}(\mathcal{S}^*_w)$$

The proof of this lemma is also in Appendix B. This lemma concludes our sequence of comparisons. By putting together what we proved so far, we have

► **Proposition 15.** *The spanner returned by DISTRIBUTED-SPANNER has a weight of $\mathcal{O}(1)\mathbf{w}(MST)$.*

Proof. By Corollary 12, Lemma 13, and Lemma 14,

$$\mathbf{w}(\mathcal{S}_w) = \mathcal{O}(1)\mathbf{w}(\mathcal{S}^*_w)$$

Summing up together these inequalities for $w \in I$,

$$\mathbf{w}(\text{output}) = \mathcal{O}(1) \sum_{w \in I} \mathbf{w}(\mathcal{S}^*_w)$$

But we recall that every vertex, and hence every edge of $\mathcal{S}^*$, is repeated $\mathcal{O}(1)$ times in the summation above, so

$$\mathbf{w}(\text{output}) = \mathcal{O}(1)\mathbf{w}(\mathcal{S}^*) = \mathcal{O}(1)\mathbf{w}(MST(G))$$ ◄

Therefore we have all the ingredients to prove Theorem 16.

► **Theorem 16** (Distributed Spanner)**.** *Given a weighted unit ball graph G with n vertices in a metric of bounded doubling dimension and a constant $\epsilon > 0$, the algorithm DISTRIBUTED-SPANNER(G,ϵ) runs in $\mathcal{O}(\log^* n)$ rounds of communication in the LOCAL model of computation, and returns a $(1+\epsilon)$-spanner of G that has constant bounds on its lightness and maximum degree. These constant bounds only depend on ϵ and the doubling dimension.*

Proof. It directly follows from Lemma 9, Lemma 10, Lemma 11, and Proposition 15. ◄

5 Adjustments for the CONGEST Model

In this section we study the problem of finding a bounded-degree $(1+\epsilon)$-spanner in the CONGEST model of computation, for a point set that is located in a doubling metric space. In the CONGEST model, every node can send a message of bounded size to every other node in a single round of communication. This makes it hard to gather any global information about the graph.

The maximal independent set algorithm of [35] still works in $\mathcal{O}(\log^* n)$ rounds of communication in the CONGEST model. But our proposed distributed algorithm (Algorithm 3) needs to gather 2-hop neighborhood information of every center in the MIS, which requires $\mathcal{O}(D)$ rounds in the CONGEST model, where D is the maximum degree of a vertex in the unit ball graph. The rest of the algorithm is performed locally and the number edges sent to every neighbor in the end is bounded by a constant, so the remaining of the algorithm only requires a constant number of rounds.

It is natural to ask whether our algorithm can be adapted to the CONGEST model, and if it requires more communication rounds compared to the LOCAL model. In this section we show how to modify our algorithm to work in the CONGEST model, and surprisingly, have no asymptotic change on its number of communication rounds.

As we mentioned earlier, the only step of our algorithm that requires more than constant rounds of communication is the aggregation of the 2-hop neighborhood information for every center in the MIS. We passed the 2-hop neighborhoods to our centralized algorithm to find an asymptotically optimal spanner on them, which was later distributed among the vertices in the neighborhood to form the final spanner. Here, for our construction in the CONGEST model, we directly address the problem of finding an asymptotically optimal spanner for the 2-hop neighborhoods, without the need to access all of the points in those neighborhoods.

Let $w \in I$ be a center in the maximal independent set. We partition the edges of the UBG on $\mathcal{N}^2(w)$ into two sets, depending on whether their length is larger than $1/2$ or not. We aim to find asymptotically optimal spanners for each partition separately. We use the notation $G_{\leq\alpha}$ to refer to the subgraph of the unit ball graph that consists of edges of length at most α. We similarly define $G_{>\alpha}$. Therefore, we can refer to the subgraphs induced by the two partitions by $G_{\leq 1/2}$ and $G_{>1/2}$.

First, we show that in constant rounds of communication, we can find a covering of the points in $\mathcal{N}^2(w)$ with at most a constant number of balls of radius $1/2$. The existence of such covering trivially follows from the definition of a doubling metric space, but finding such covering in the distributed setting is not trivial. Therefore, we introduce the following procedure: Every center $v \in \mathcal{N}^1(w)$ (including w itself) finds a maximal independent set $I_{1/4}(v)$ of the vertices $\mathcal{N}^1(v)$ in $G_{\leq 1/4}$, and sends it to w, all centers at the same time. Recall that $\mathcal{N}^1(v)$ is the set of neighbors of v in the UBG, and a maximal independent set in $G_{\leq 1/4}$ is simply a maximal set of vertices where the pair-wise distance of each two vertex is at least $1/4$. Finding this maximal independent set for each v can be easily done using a (centralized) greedy algorithm, and the size of such maximal independent set would be bounded by a constant according to the packing lemma. Therefore, this step can be done in constant number of rounds. Afterwards, w calculates a maximal independent set $\mathcal{I}(w)$ of the vertices $\cup_{v\in\mathcal{N}^1(w)} I_{1/4}(v)$ in $G_{\leq 1/4}$. We show that the centers in $\mathcal{I}$ satisfy our desired properties.

▶ **Lemma 17.** *The union of the balls of radius $1/2$ around the centers in $\mathcal{I}(w)$ cover $\mathcal{N}^2(w)$. Furthermore, the size of $\mathcal{I}(w)$ is bounded by a constant.*

The proof of this lemma is moved to Appendix C. Next, every center $v \in \mathcal{I}(w)$ calculates a $(1+\epsilon)$-spanner $\mathcal{S}_{\leq 1/2}(v)$ of the point set $\mathcal{N}^1(v)$ using the centralized algorithm, and notifies its neighbors about their connections. We prove that the union of these spanners, would be a spanner for one of the partitions, i.e. the edges of length at most $1/2$ in $\mathcal{N}^2(w)$. The pseudo-code of this procedure is available in Algorithm 5

▶ **Lemma 18.** *The union of the spanners $\mathcal{S}_{\leq 1/2}(v)$ for $v \in \mathcal{I}(w)$ is a $(1+\epsilon)$-spanner of $\mathcal{N}^2(w)$ in $G_{\leq 1/2}$. The maximum degree and the lightness of this spanner are both bounded by constants.*

The proof of this lemma can be found in Appendix C.

Algorithm 4 The CONGEST spanner algorithm.

Input. A unit ball graph $G(V,E)$ in a metric with doubling dimension d and an $\epsilon > 0$.
Output. A light-weight bounded-degree $(1+\epsilon)$-spanner of G.

procedure CONGEST-SPANNER(G, ϵ)
    Find a maximal independent set I of G using [35]
    Run SPAN-SHORT-EDGES on the vertices of G
    Run SPAN-LONG-EDGES on the vertices of G

Algorithm 5 Finding a spanner of the edges of length smaller than $1/2$ in $\mathcal{N}^2(w)$.

function SPAN-SHORT-EDGES(vertex u)
    if $u \in I$ **then**
        Send a signal of type 1 to every $v \in \mathcal{N}^1(u)$.
        Wait for their maximal independent sets, $I_{1/4}(v)$s.
        Calculate a maximal independent set of $\cup_{v \in \mathcal{N}^1(u)} I_{1/4}(v)$ in $G_{<1/4}$ greedily.
        Store this maximal independent set in $\mathcal{I}(u)$.
        Send a signal of type 2 to every $v \in \mathcal{I}(u)$.
    if received type 1 signal from some w **then**
        Calculate a maximal independent set of $\mathcal{N}^1(w)$ in $G_{1/4}$ greedily.
        Send this maximal independent set to w.
    if received type 2 signal from some w **then**
        Calculate $\mathcal{S}_{\leq 1/2}(u) \leftarrow$ CENTRALIZED-SPANNER($\mathcal{N}^1(u)$, ϵ)
        for $e = (a,b)$ in $\mathcal{S}_{\leq 1/2}(u)$ **do**
            Send e to a and b
    Receive and store the edges sent by other centers

Now we find a spanner for the other partition, the edges of length larger than $1/2$ in $\mathcal{N}^2(w)$. The procedure is as follows: First, every center $v \in \mathcal{N}^1(w)$ calculates a maximal independent set $I_{\epsilon/40}(v)$ of $\mathcal{N}^1(v)$ in $G_{\leq \epsilon/40}$ and sends it to w. Again, the size of each maximal independent set is $\mathcal{O}(\epsilon^{-d})$ by the packing lemma, which is constant. Therefore, this step takes only constant number of rounds. Afterwards, w finds a maximal independent set $\mathcal{I}'(w)$ of $\cup_{v \in \mathcal{N}^1(w)} I_{\epsilon/40}(v)$ in $G_{\leq \epsilon/40}$. Then w constructs a $(1+\epsilon/5)$-spanner $\mathcal{S}'(w)$ of $\mathcal{I}'(w)$ in G using the centralized algorithm, and announces the edges of the spanner to their corresponding endpoints. Finally, every center $v \in \mathcal{I}'(w)$ calculates a $(1+\epsilon)$-spanner $\mathcal{S}''(v)$ of its $\epsilon/20$ neighborhood and announces its edges to their endpoints. We show that the union of $S'(w)$ and $S''(v)$s for $v \in \mathcal{I}'(w)$ would form a $(1+\epsilon)$-spanner of the second partition, i.e. the edges of larger than $1/2$. The pseudo-code of this procedure is available in Algorithm 6

▶ **Lemma 19.** *The union of the balls of radius $\epsilon/20$ around the centers in $\mathcal{I}'(w)$ cover $\mathcal{N}^2(w)$. Furthermore, the size of $\mathcal{I}'(w)$ is bounded by a constant.*

Proof. Similar to the proof of Lemma 17. ◀

▶ **Lemma 20.** *The union of the spanners $\mathcal{S}'(w)$ and $\mathcal{S}''(v)$ for $v \in \mathcal{I}'(w)$ forms a $(1+\epsilon)$-spanner of $\mathcal{N}^2(w)$ in $G_{>1/2}$. The maximum degree and the lightness of this spanner are btoh bounded by constants.*

The proof of this lemma is also in Appendix C.

Algorithm 6 Finding a spanner of the edges of length larger than $1/2$ in $\mathcal{N}^2(w)$.

function SPAN-LONG-EDGES(vertex u)
  if $u \in I$ **then**
    Send a signal of type 3 to every $v \in \mathcal{N}^1(u)$.
    Wait for their maximal independent sets, $I_{\epsilon/40}(v)$s.
    Calculate a maximal independent set of $\cup_{v \in \mathcal{N}^1(u)} I_{\epsilon/40}(v)$ in $G_{<\epsilon/40}$ greedily.
    Store this maximal independent set in $\mathcal{I}'(u)$.
    Send a signal of type 4 to every $v \in \mathcal{I}'(u)$.
    Calculate $\mathcal{S}'(u) \leftarrow$ CENTRALIZED-SPANNER($\mathcal{I}'(u)$, $\epsilon/5$)
    for $e = (a, b)$ in $\mathcal{S}'(u)$ **do**
      Send e to a and b
  if received type 3 signal from some w **then**
    Calculate a maximal independent set of $\mathcal{N}^1(w)$ in $G_{\epsilon/40}$ greedily.
    Send this maximal independent set to w.
  if received type 4 signal from some w **then**
    Let $\mathcal{N}^{\epsilon/20}(u)$ be the $\epsilon/20$ neighborhood of u, i.e. the set of vertices that are at distance $\epsilon/20$ or less from u.
    Calculate $\mathcal{S}''(u) \leftarrow$ CENTRALIZED-SPANNER($\mathcal{N}^{\epsilon/20}(u)$, ϵ)
    for $e = (a, b)$ in $\mathcal{S}''(u)$ **do**
      Send e to a and b
  Receive and store the edges sent by other centers

The union of the two spanners for the two partitions form a spanner for the 2-hop neighborhood of w, the goal we wanted to achieve in the CONGEST model. This completes our adjustments in this model.

▶ **Theorem 21** (CONGEST Spanner)**.** *Given a weighted unit ball graph G with n vertices in a metric of bounded doubling dimension and a constant $\epsilon > 0$, the algorithm CONGEST-SPANNER(G,ϵ) runs in $\mathcal{O}(\log^* n)$ rounds of communication in the CONGEST model of computation, and returns a $(1+\epsilon)$-spanner of G that has constant bounds on its lightness and maximum degree. These constant bounds only depend on ϵ and the doubling dimension.*

Proof. The proof follows from Lemma 18 and Lemma 20. ◀

We defer our low-intersection construction as well as our experimental results to the full version of the paper due to space limitation.

6 Conclusions

In this paper we resolved an open question from 2006 and we proved the existence of light-weight bounded-degree $(1+\epsilon)$-spanners for unit ball graphs in the spaces of bounded doubling dimension. Moreover, we provided a centralized construction and a distributed construction in the LOCAL model that finds a spanner with these properties. Our distributed construction runs in $\mathcal{O}(\log^* n)$ rounds, where n is the number of vertices in the graph. If a maximal independent set of the unit ball graph is known beforehand, our algorithm runs in constant number of rounds. Next, we showed how to adjust our distributed construction to work in the CONGEST model, without touching its asymptotic round complexity. In this way, we provided the first CONGEST algorithm for finding a light spanner of unit ball graphs.

In the full version, we further adjusted these algorithms for the case of unit disk graphs in the two dimensional Euclidean plane, and we presented the first centralized and distributed constructions for a light-weight bounded-degree $(1+\epsilon)$-spanner that also has a linear number of edge intersections in total. This can be useful for practical purposes if minimizing the number of edge intersections is a priority. We proved, based on this low-intersection property, that our spanner has sub-linear separators, and a separator hierarchy, and we were able to generalize this result to higher dimensions of Euclidean spaces.

Finally, we performed experiments (in the full version) on random point sets in the two dimensional Euclidean plane, to ensure that our theoretical bounds are also supported by enough empirical evidence. Our results show that our construction performs efficiently with respect to the maximum degree, size, and total weight.

References

1. Mohammad Ali Abam, Mark De Berg, Mohammad Farshi, and Joachim Gudmundsson. Region-fault tolerant geometric spanners. *Discrete & Computational Geometry*, 41(4):556–582, 2009. doi:10.1007/s00454-009-9137-7.
2. Ingo Althöfer, Gautam Das, David Dobkin, and Deborah Joseph. Generating sparse spanners for weighted graphs. In John R. Gilbert and Rolf Karlsson, editors, *Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 447 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 1990. doi:10.1007/3-540-52846-6_75.
3. Khaled Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(4):408–421, 2003. doi:10.1109/TPDS.2003.1195412.
4. Baruch Awerbuch, Bonnie Berger, Lenore Cowen, and David Peleg. Near-linear time construction of sparse neighborhood covers. *SIAM Journal on Computing*, 28(1):263–277, 1998. doi:10.1137/S0097539794271898.
5. Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. Additive spanners and (α, β)-spanners. *ACM Transactions on Algorithms*, 7(1):5, 2010. doi:10.1145/1868237.1868242.
6. Sujoy Bhore, Arnold Filtser, Hadi Khodabandeh, and Csaba D Tóth. Online spanners in metric spaces. *arXiv preprint*, 2022. arXiv:2202.09991.
7. Glencora Borradaile, Hung Le, and Christian Wulff-Nilsen. Greedy spanners are optimal in doubling metrics. In *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2371–2379. Society for Industrial and Applied Mathematics, 2019. doi:10.1137/1.9781611975482.145.
8. Rebecca Braynard, Dejan Kostic, Adolfo Rodriguez, Jeffrey Chase, and Amin Vahdat. Opus: an overlay peer utility service. In *Proceedings of the 5th IEEE Conference on Open Architectures and Network Programming (OPENARCH)*, pages 167–178. IEEE, 2002. doi:10.1109/OPNARC.2002.1019237.

9 T-H Hubert Chan and Anupam Gupta. Small hop-diameter sparse spanners for doubling metrics. *Discrete & Computational Geometry*, 41(1):28–44, 2009.

10 Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. Fault tolerant spanners for general graphs. *SIAM Journal on Computing*, 39(7):3403–3423, 2010. doi:10.1137/090758039.

11 Paul Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989. doi:10.1016/0022-0000(89)90044-5.

12 Edith Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. *SIAM Journal on Computing*, 28(1):210–236, 1998. doi:10.1137/S0097539794261295.

13 Mirela Damian, Saurav Pandit, and Sriram Pemmaraju. Distributed spanner construction in doubling metric spaces. In *International Conference on Principles of Distributed Systems*, pages 157–171. Springer, 2006.

14 Mirela Damian, Saurav Pandit, and Sriram Pemmaraju. Local approximation schemes for topology control. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 208–217, 2006.

15 Gautam Das. The visibility graph contains a bounded-degree spanner. In *Proceedings of the 9th Canadian Conference on Computational Geometry (CCCG)*, pages 70–75, 1997. URL: https://cccg.ca/proceedings/1997/.

16 Andrew Dobson and Kostas E. Bekris. Sparse roadmap spanners for asymptotically near-optimal motion planning. *International Journal of Robotics Research*, 33(1):18–47, 2014. doi:10.1177/0278364913498292.

17 Michael Elkin. Computing almost shortest paths. *ACM Transactions on Algorithms*, 1(2):283–323, 2005. doi:10.1145/1103963.1103968.

18 Michael Elkin, Arnold Filtser, and Ofer Neiman. Distributed construction of light networks. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 483–492, 2020.

19 Michael Elkin and David Peleg. $(1+\varepsilon,\beta)$-spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004. doi:10.1137/S0097539701393384.

20 Michael Elkin and Jian Zhang. Efficient algorithms for constructing $(1+\varepsilon,\beta)$-spanners in the distributed and streaming models. *Distributed Computing*, 18(5):375–385, 2006. doi:10.1007/s00446-005-0147-2.

21 David Eppstein. Spanning trees and spanners. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. North-Holland, 2000. doi:10.1016/B978-044482537-7/50010-3.

22 David Eppstein and Siddharth Gupta. Crossing patterns in nonplanar road networks. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–9, 2017.

23 David Eppstein and Hadi Khodabandeh. On the edge crossings of the greedy spanner. In *37th International Symposium on Computational Geometry*, volume 12, page 37, 2021.

24 David Eppstein and Hadi Khodabandeh. Optimal spanners for unit ball graphs in doubling metrics. *arXiv preprint*, 2021. arXiv:2106.15234.

25 Mohammad Farshi and Joachim Gudmundsson. Experimental study of geometric t-spanners. *ACM Journal of Experimental Algorithmics*, 14:1.3:1–1.3:29, 2009. doi:10.1145/1498698.1564499.

26 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the streaming model: the value of space. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 745–754. Society for Industrial and Applied Mathematics, 2005.

27 Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 9–17, 2016.

28 Jie Gao, Leonidas J Guibas, John Hershberger, Li Zhang, and An Zhu. Geometric spanners for routing in mobile networks. *IEEE journal on selected areas in communications*, 23(1):174–185, 2005.

29 Lee-Ad Gottlieb. A light metric spanner. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 759–772. IEEE, 2015. doi:10.1109/FOCS.2015.52.

30 Joachim Gudmundsson, Christos Levcopoulos, and Giri Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM Journal on Computing*, 31(5):1479–1500, 2002.

31 Jonathan P Jenkins, Iyad A Kanj, Ge Xia, and Fenghui Zhang. Local construction of spanners in the 3d space. *IEEE Transactions on Mobile Computing*, 11(7):1140–1150, 2012.

32 Lujun Jia, Rajmohan Rajaraman, and Christian Scheideler. On local algorithms for topology control and routing in ad hoc networks. In *Proceedings of the 15th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 220–229. ACM, 2003. doi:10.1145/777412.777447.

33 Iyad A Kanj, Ljubomir Perković, and Ge Xia. Computing lightweight spanners locally. In *International Symposium on Distributed Computing*, pages 365–378. Springer, 2008.

34 J. Mark Keil. Approximating the complete Euclidean graph. In Rolf Karlsson and Andrzej Lingas, editors, *Proceedings of the 1st Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 318 of *Lecture Notes in Computer Science*, pages 208–213. Springer, 1988. doi:10.1007/3-540-19487-8_23.

35 Fabian Kuhn, Thomas Moscibroda, and Rogert Wattenhofer. On the locality of bounded growth. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 60–68, 2005.

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

37 Ning Li, Jennifer C Hou, and Lui Sha. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, 2005.

38 Xiang-Yang Li, Gruia Calinescu, Peng-Jun Wan, and Yu Wang. Localized delaunay triangulation with application in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.

39 Xiang-Yang Li, Peng-Jun Wan, and Yu Wang. Power efficient and sparse spanner for wireless ad hoc networks. In *Proceedings Tenth International Conference on Computer Communications and Networks (Cat. No. 01EX495)*, pages 564–567. IEEE, 2001.

40 James D. Marble and Kostas E. Bekris. Asymptotically near-optimal planning with probabilistic roadmap spanners. *IEEE Transactions on Robotics*, 29(2):432–444, 2013. doi:10.1109/TRO.2012.2234312.

41 Giri Narasimhan and Michiel Smid. *Geometric spanner networks*. Cambridge University Press, 2007.

42 David Peleg and Liam Roditty. Localized spanner construction for ad hoc networks with variable transmission range. *ACM Transactions on Sensor Networks (TOSN)*, 7(3):1–14, 2010.

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

44 Liam Roditty and Uri Zwick. On dynamic shortest paths problems. *Algorithmica*, 61(2):389–401, 2011. doi:10.1007/s00453-010-9401-5.

45 Johannes Schneider and Roger Wattenhofer. A log-star distributed maximal independent set algorithm for growth-bounded graphs. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 35–44, 2008.

46 Michiel Smid. The weak gap property in metric spaces of bounded doubling dimension. In *Efficient Algorithms*, pages 275–289. Springer, 2009.

47 Wenjie Wang, Cheng Jin, and Sugih Jamin. Network overlay construction under limited end-to-end reachability. In *Proceedings of the 24th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 2124–2134. IEEE, 2005.

A Omitted Proofs from Section 3

▶ **Lemma 5.** *The spanner returned by* Centralized-Spanner *has a stretch factor of* $1+\epsilon$.

Proof. We recall that the output of Approximate-Greedy is a light-weight bounded-degree $(1+\epsilon')$-spanner of the complete weighted graph on the point set. So an edge e of length $|e| > 1+\epsilon'$ cannot be used to approximate any edges in the UBG, i.e. if $(x, y) \in E$ then e cannot belong to the shortest path between x and y in S; otherwise the length of the path would exceed $1+\epsilon'$ which cannot happen. So we may safely remove these edges in the first step of the refinement procedure without replacing them.

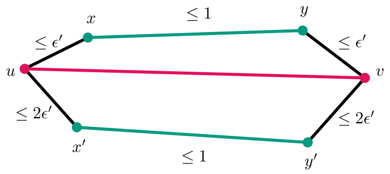


Figure 3 An edge (x, y) of the UBG that uses a longer than unit length edge (u, v) of the spanner on its shortest path, which is then replaced by (x', y') during the replacement procedure.

Also, any edge of length in the range $(1, 1+\epsilon']$ that is not used in a shortest path between any two endpoints of an edge of UBG can be removed as well, because removing them does not change the stretch-factor of the spanner. Now consider an edge $(x, y) \in E$ of the UBG that uses a spanner edge $e = (u, v) \in S$ that $|e| \in (1, 1+\epsilon']$ on its shortest path. We want to prove that after the replacement of e, the shortest path between x and y remains within $1+\epsilon$ factor of their distance. Clearly, we have $\|ux\| \leq \epsilon'$ and $\|vy\| \leq \epsilon'$; otherwise the length of the path $xuvy$ would be more than $1+\epsilon'$, contradicting the fact that it is approximating an edge of length at most 1. This shows that (x, y) would be a valid replacement edge for e. So we can safely assume that the algorithm finds a (possible weak) replacement edge $(x', y') \in E$ for e (Figure 3). This replacement edge might be a normal replacement edge or a weak replacement edge. Either way, we have $\|ux'\| \leq 2\epsilon'$ and $\|vy'\| \leq 2\epsilon'$. By the triangle inequality

$$\|x'x\| \leq \|x'u\| + \|ux\| \leq 2\epsilon' + \epsilon' = 3\epsilon'$$

Similarly, $\|yy'\| \leq 3\epsilon'$. Therefore

$$\|x'y'\| \leq \|x'x\| + \|xy\| + \|yy'\| \leq \|xy\| + 6\epsilon' \tag{1}$$

Denote the shortest spanner path between x and x' by $P_{xx'}$ and similarly define $P_{yy'}$. Consider the spanner path $P = P_{xx'}x'y'P_{y'y}$ that connects x and y. Using Equation 1 the length of the path P is

$$|P| = |P_{xx'}| + \|x'y'\| + |P_{y'y}| \leq \|xy\| + 6\epsilon' + |P_{xx'}| + |P_{y'y}| \tag{2}$$

The changes that we make in the refinement process do not affect the length of short paths like $P_{xx'}$ and $P_{y'y}$. So we have

$$|P_{xx'}| \leq (1+\epsilon')\|xx'\| \leq 3\epsilon'(1+\epsilon')$$

Similarly, $|P_{yy'}| \leq 3\epsilon'(1+\epsilon')$. Putting these into Equation 2 and using $\epsilon = 36\epsilon'$,

$$|P| \leq \|xy\| + 6\epsilon' + 6(1+\epsilon')\epsilon' \leq \|xy\| + \frac{\epsilon}{1+\epsilon'} \tag{3}$$

But since e was previously approximating the edge (x, y), we know that $(1+\epsilon')\|xy\| \geq |e| > 1$ or equivalently $\|xy\| > 1/(1+\epsilon')$. Substituting this into Equation 3,

$$|P| \leq \|xy\| + \epsilon\|xy\| = (1+\epsilon)\|xy\|$$

So S is a $(1+\epsilon)$-spanner of G. ◀

▶ **Lemma 6.** *The spanner returned by* Centralized-Spanner *has a weight of* $\mathcal{O}(1)\mathbf{w}(MST)$.

Proof. Again, we use the fact that the output of Approximate-Greedy has weight $\mathcal{O}(1)\mathbf{w}(MST(G))$. During the refinement process, every edge is replaced by an edge of smaller length, so the whole weight of the graph does not increase during the refinement process. Therefore in the end $\mathbf{w}(S) = \mathcal{O}(1)\mathbf{w}(MST(G))$. ◀

▶ **Lemma 7.** *The spanner returned by* Centralized-Spanner *has bounded degree.*

Proof. It is clear from the algorithm that immediately after processing an edge $e = (u, v)$, the degree of u and v does not increase; it may decrease due to the removal of the edge which is fine. But if a replacement edge (x, y) is added after the removal of e then the degree of x and y is increased by at most 1. We need to make sure this increment is bounded for every vertex.

Let x be an arbitrary vertex of G and let (x, y) and (x, z) be two replacement edges that have been added to x in this order as a result of the refinement process. We claim that $\|yz\| > \epsilon'$ holds. Assume, on the contrary, that $\|yz\| \leq \epsilon'$, and also assume that (x, z) has been added in order to replace an edge (u, v) of the spanner. Then by the triangle inequality

$$\|vy\| \leq \|vz\| + \|zy\| \leq 2\epsilon'$$

Also $\|ux\| \leq \epsilon' < 2\epsilon'$ because (x, z) is added to replace (u, v). But the last two inequalities contradict the fact that (x, y) cannot be a weak replacement for (u, v).

Now that we have proved $\|yz\| > \epsilon'$ we can use the packing property of the bounded doubling dimension to bound the number of such replacement edges around x. All the other endpoints of such replacement edges are included in ball of radius 1 around x, and the distance between every two such points is at least ϵ'. Thus by the packing property there can be at most $(\frac{4}{\epsilon'})^d = \epsilon^{-\mathcal{O}(d)}$ many replacement edges incident to x. ◀

B Omitted Proofs from Section 4

▶ **Lemma 9.** Distributed-Spanner *can be done in* $\mathcal{O}(\log^* n)$ *rounds of communication.*

Proof. The pre-processing step of finding the maximal independent set takes $\mathcal{O}(\log^* n)$ rounds of communication [35]. Retrieving the 2-hop neighborhood information can be done in $\mathcal{O}(1)$ rounds of communication. Computing the greedy spanner is done locally, and the edges are sent to their endpoints, which again can be done in $\mathcal{O}(1)$ rounds of communication. Overall, the algorithm requires $\mathcal{O}(\log^* n)$ rounds of communication. ◀

▶ **Lemma 10.** *The spanner returned by* Distributed-Spanner *has a stretch factor of* $1+\epsilon$.

Proof. From section 3 we know that $\mathcal{S}_w$ is a light-weight bounded-degree $(1+\epsilon)$-spanner of $\mathcal{N}^2(w)$. Let $u, v \in V$ be chosen arbitrarily. We need to make sure there is a path of length at most $(1+\epsilon)d_G(u, v)$ between u and v in the output.

First we prove this for the case that $(u, v) \in E$. So let $e = (u, v) \in E$. Then u is either in I or has a neighbor in I, according to choice of I. In any case, the edge e belongs to $\mathcal{N}^2(w)$ for some $w \in I$, which means that there is a path $P \subset \mathcal{S}_w$ of length at most $(1+\epsilon)|e|$

that connects u and v. The edges of P are all included in the final spanner according to the algorithm, so the output includes this path between u and v, which has a length at most $(1+\epsilon)|e|$ and so the distance inequality is satisfied.

If $(u,v) \notin E$, we can take the shortest path $u = p_0, p_1, \cdots, p_k = v$ between them in G and append the corresponding $(1+\epsilon)$-approximate paths $P_0, P_1, \cdots, P_{k-1}$ of the edges $p_0p_1, p_1p_2, \cdots, p_{k-1}p_k$, respectively, to get a $(1+\epsilon)$-approximate path for $p_0p_1 \cdots p_k$. This implies that the stretch factor of the output is indeed $1+\epsilon$. ◀

▶ **Lemma 11.** *The spanner returned by* DISTRIBUTED-SPANNER *has a bounded degree.*

Proof. First we use the packing lemma to prove that any vertex $v \in V$ appears at most a constant number of times in different neighborhoods, $\mathcal{N}^2(w)$ for $w \in I$. Because $v \in \mathcal{N}^2(w)$ implies that $\|vw\| \leq 2$, any vertex $w \in I$ such that $v \in \mathcal{N}^2(w)$ should be contained in the ball of radius 2 around v. But all such ws are chosen from I, which is an independent set of G, so the distance between every two such vertex is at least 1. By the packing property, the maximum number of such vertices would be $8^d = \mathcal{O}(1)$.

Now that every vertex appears in at most in 8^d different sets $\mathcal{N}^2(w)$, for $w \in I$, and from section 3 we already knew that every vertex has bounded degree in any of $\mathcal{S}_w$s, it immediately follows that every vertex has bounded degree in the final spanner. ◀

▶ **Lemma 13.** *Define $\mathcal{T}$ to be the optimal Steiner tree on the set of vertices $\mathcal{N}^3(w)$, where only vertices in $\mathcal{N}^2(w)$ are required and the rest of them are optional. Then*

$$\mathbf{w}(MST(\mathcal{N}^2(w))) \leq 2\mathbf{w}(\mathcal{T})$$

Proof. This is a well-known fact that implies a 2-approximation for minimum Steiner tree problem. The idea is if we run a full DFS on the vertices of $\mathcal{T}$ and we write every vertex once we open and once we close it, then we get a cycle whose shortcut on optional edges will form a path on the required vertices. The weight of the cycle is at least $\mathbf{w}(MST(\mathcal{N}^2(w)))$ and at most $2\mathbf{w}(\mathcal{T})$, which proves the result. ◀

▶ **Lemma 14.** *Let $\mathcal{S}^*$ be the output of* CENTRALIZED-SPANNER*(G, ϵ) and let $\mathcal{S}^*_w$ be the induced subgraph of $\mathcal{S}^*$ on $\mathcal{N}^3(w)$. Then*

$$\mathbf{w}(\mathcal{T}) \leq \mathbf{w}(\mathcal{S}^*_w)$$

Proof. We prove that for $\epsilon \leq 1$, $\mathcal{S}^*_w$ forms a forest that connects all the vertices in $\mathcal{N}^2(w)$ in a single component. So $\mathcal{S}^*_w$ is a feasible solution to the minimum Steiner tree problem on the set of vertices $\mathcal{N}^3(w)$ with required vertices being $\mathcal{N}^2(w)$. Thus $\mathbf{w}(\mathcal{T}) \leq \mathbf{w}(\mathcal{S}^*_w)$.

Now we just need to prove that the vertices in $\mathcal{N}^2(w)$ are connected in $\mathcal{S}^*_w$. Let u be an i-hop neighbor of w and v be an $i+1$-hop neighbor of w for some $w \in I$ and $i = 0, 1$. Assume that $(u,v) \in E$. It is enough to prove that u and v are connected in $\mathcal{S}^*_w$. In order to do so, we observe that there is a path of length at most $(1+\epsilon)\|uv\|$ between u and v in $\mathcal{S}^*$. We show that this path is contained in $\mathcal{N}^3(w)$ and we complete the proof in this way, because $\mathbf{w}(\mathcal{S}^*_w)$ is nothing but the induced subgraph of $\mathcal{S}^*$ on $\mathcal{N}^3(w)$.

Assume, on the contrary, that there is a vertex $x \notin \mathcal{N}^3(w)$ on the $(1+\epsilon)$-path between u and v. This means that x is not a 1-hop neighbor of any of u and v, because otherwise x would have been in $\mathcal{N}^3(w)$. So $\|ux\| > 1$ and $\|vx\| > 1$. Thus the length of the path would be at least $\|ux\| + \|xv\| > 2 \geq (1+\epsilon) \geq (1+\epsilon)\|uv\|$ which is a contradiction. ◀

C Omitted Proofs from Section 5

▶ **Lemma 17.** *The union of the balls of radius* $1/2$ *around the centers in* $\mathcal{I}(w)$ *cover* $\mathcal{N}^2(w)$*. Furthermore, the size of* $\mathcal{I}(w)$ *is bounded by a constant.*

Proof. Let $v \in \mathcal{N}^2(w)$ be an arbitrary point. Thus there exists $u \in \mathcal{N}^1(w)$ that $v \in \mathcal{N}^1(u)$. Let $I_{1/4}(u)$ be the maximal independent set of the vertices $\mathcal{N}^1(u)$ in $G_{\leq 1/4}$, that u calculates and sends to w in the first step. There exists $v' \in I_{1/4}(u)$ that $\|vv'\| \leq 1/4$. Similarly, there exists $v'' \in \mathcal{I}(w)$ that $\|v'v''\| \leq 1/4$. By the triangle inequality, $\|vv''\| \leq 1/2$, i.e. v is covered by a ball of radius $1/2$ around v''.

On the other hand, $\mathcal{I}(w)$ is contained in a ball of radius 2 and every pair of points in $\mathcal{I}(w)$ have a distance of at least $1/4$. Thus, by the packing lemma, he size of $\mathcal{I}(w)$ is bounded by a constant. ◀

▶ **Lemma 18.** *The union of the spanners* $\mathcal{S}_{\leq 1/2}(v)$ *for* $v \in \mathcal{I}(w)$ *is a* $(1+\epsilon)$*-spanner of* $\mathcal{N}^2(w)$ *in* $G_{\leq 1/2}$*. The maximum degree and the lightness of this spanner are both bounded by constants.*

Proof. First, we prove the $1+\epsilon$ stretch-factor. Let (u, v) be a pair in $\mathcal{N}^2(w)$ such that $\|uv\| \leq 1/2$. By Lemma 17 we know there exists $u' \in \mathcal{I}(w)$ that $\|uu'\| \leq 1/2$. Thus $\|vu'\| \leq 1$ which means that $u, v \in \mathcal{N}^1(u')$ and there would be a $(1+\epsilon)$-path for this pair in $\mathcal{S}_{\leq 1/2}(u')$, which would be present in the union of the spanners.

The degree bound follows from the fact that, by the packing lemma, every point in $\mathcal{N}^2(w)$ is appeared in at most a constant number of one-hop neighborhoods and therefore in at most a constant number of spanners constructed the elements in $\mathcal{I}(w)$. Since in every spanner it has a bounded degree, in the union it will have a bounded degree as well.

To prove the lightness bound we follow a similar approach to the proof of Proposition 15. The weight of each spanner $\mathcal{S}_{\leq 1/2}(v)$ is $\mathcal{O}(1)\mathbf{w}(MST(\mathcal{N}^1(v)))$. The weight of the MST is at most twice the weight of the optimal Steiner tree on $\mathcal{N}^2(v)$ with the required vertices being $\mathcal{N}^1(v)$. And the weight of this optimal Steiner tree is at most equal to the weight of the induced sub-graph of an (asymptotically) optimal $(1+\epsilon)$-spanner of G on the subset of vertices $\mathcal{N}^2(v)$. Summing up these subgraphs for different vs and different ws would end up with adding at most a constant factor to the weight of the optimal spanner, which proves that the lightness would be bounded by a constant. ◀

▶ **Lemma 20.** *The union of the spanners* $\mathcal{S}'(w)$ *and* $\mathcal{S}''(v)$ *for* $v \in \mathcal{I}'(w)$ *forms a* $(1+\epsilon)$*-spanner of* $\mathcal{N}^2(w)$ *in* $G_{>1/2}$*. The maximum degree and the lightness of this spanner are btoh bounded by constants.*

Proof. Again, we first prove the $1+\epsilon$ stretch-factor of the spanner. Let (u, v) be a pair in $\mathcal{N}^2(w)$ that $\|uv\| > 1/2$. Let u' and v' be centers in $\mathcal{I}'(w)$ that are at distance of at most $\epsilon/20$ from u and v, respectively. Such centers exist according to Lemma 19. Consider the $(1+\epsilon)$-path connecting u to u' in $\mathcal{S}''(u')$ and the $(1+\epsilon)$-path connecting v to v' in $\mathcal{S}''(v')$. We can attach these paths together with the $(1+\epsilon/5)$-path between u' and v' in $\mathcal{S}'(w)$ to get a path between u and v. The stretch of this path would be at most

$$\frac{(1+\epsilon)(\|uu'\| + \|vv'\|) + (1+\epsilon/5)\|u'v'\|}{\|uv\|} = \frac{(1+\epsilon)(\|uu'\| + \|vv'\|)}{\|uv\|} + \frac{(1+\epsilon/5)\|u'v'\|}{\|uv\|}$$

But,

$$\frac{(1+\epsilon)(\|uu'\| + \|vv'\|)}{\|uv\|} \leq \frac{(1+\epsilon)\epsilon/10}{1/2} \leq \frac{2\epsilon}{5}$$

Also,

$$\frac{(1+\epsilon/5)\|u'v'\|}{\|uv\|} \leq \frac{(1+\epsilon/5)(\|uv\|+\|uu'\|+\|vv'\|)}{\|uv\|} \leq 1+\epsilon/5+\frac{(1+\epsilon/5)\epsilon/10}{\|uv\|}$$

Bounding the last term,

$$\frac{(1+\epsilon/5)\epsilon/10}{\|uv\|} \leq \frac{(6/5)\epsilon/10}{1/2} = \frac{6\epsilon}{25}$$

Therefore, the stretch of the uv-path would be upper bounded by,

$$\frac{2\epsilon}{5}+1+\epsilon/5+\frac{6\epsilon}{25} < 1+\epsilon$$

An approach similar to the proof of Lemma 18 shows that the degree of every vertex in the union of $\mathcal{S}''(v)$s would be bounded by a constant. We do not repeat the details of the proof here. From the properties of our centralized construction, the degree of every vertex would be bounded in $\mathcal{S}'(w)$ as well. Thus, the degree of every vertex in the union of these spanners would be bounded by a constant.

To prove the lightness bound, we bound the weight of each spanner separately. First, we bound the total weight of $\mathcal{S}'(w)$. We know from the properties of our centralized construction, that $\mathbf{w}(\mathcal{S}'(w)) = \mathcal{O}(1)\mathbf{w}(MST(\mathcal{I}'(w)))$. But $\mathbf{w}(MST(\mathcal{I}'(w))) \leq 2MST(\mathcal{N}^2(w))$, so $\mathbf{w}(\mathcal{S}'(w)) = \mathcal{O}(1)\mathbf{w}(MST(\mathcal{N}^2(w)))$. Therefore, by Lemma 13 and Lemma 14 the total weight of $\mathcal{S}'(w)$ spanners for different centers w would sum up to at most a constant factor of the weight of the optimal spanner.

Next, we bound the total weight of $\mathcal{S}''(v)$ spanners. Again, we know from the properties of our centralized construction that $\mathbf{w}(\mathcal{S}''(v)) = \mathcal{O}(1)\mathbf{w}(MST(\mathcal{N}^{\epsilon/20}(v)))$. Assuming that $\mathcal{S}^*$ is an optimal spanner on the point set, we can observe that any $(1+\epsilon)$-path (in $\mathcal{S}^*$) between any pair of vertices in $\mathcal{N}^{\epsilon/20}(v)$ must be completely contained in a ball of radius $3\epsilon/20$, otherwise the length of the path would be more than $(1+\epsilon)\epsilon/10$, the maximum allowed length for any $(1+\epsilon)$-path of any pair in the $\mathcal{N}^{\epsilon/20}(v)$ neighborhood. Therefore, the induced sub-graph of $\mathcal{S}^*$ on $\mathcal{N}^{3\epsilon/20}(v)$ has a connected component connecting the vertices of $\mathcal{N}^{\epsilon/20}(v)$. Thus, its weight is at least equal to the weight of a minimum Steiner tree on $\mathcal{N}^{3\epsilon/20}(v)$, with the required vertices being $\mathcal{N}^{\epsilon/20}(v)$. This is at least equal to $\mathbf{w}(MST(\mathcal{N}^{\epsilon/20}(v)))/2$. Therefore, the weight of $\mathcal{S}''(v)$ is bounded above by a constant factor of the weight of the induced sub-graph of $\mathcal{S}^*$ on $\mathcal{N}^{3\epsilon/20}(v)$. Summing up these bounds for every v in every w would lead to at most a constant repetitions of every vertex and every edge (similar to Proposition 15) in $\mathcal{S}^*$, which shows that the total weight of $\mathcal{S}''(v)$ for different vertices of v would be bounded by a constant factor of the weight of the optimal spanner. ◀