New Additive Spanner Lower Bounds by an Unlayered Obstacle Product

1st Greg Bodwin
EECS Department
University of Michigan
Ann Arbor, USA
bodwin@umich.edu

2nd Gary Hoppenworth
EECS Department
University of Michigan
Ann Arbor, USA
garytho@umich.edu

Abstract—For an input graph $G$, an additive spanner is a sparse subgraph $H$ whose shortest paths match those of $G$ up to small additive error. We prove two new lower bounds in the area of additive spanners:

- We construct $n$-node graphs for which any additive spanner on $O(n)$ edges must increase a pairwise distance by $\Theta(n^{1/2})$. This improves on a recent lower bound of $\Omega(n^{1/4})$ by Lu, Wein, Vassilevska Williams, and Xu [SODA ’12].
- A classic result by Coppersmith and Elkin [SODA ’05] proves that for any $n$-node graph $G$ and set of $p = O(n^{1/2})$ demand pairs, one can exactly preserve all pairwise distances among demand pairs using a spanner on $O(n)$ edges. They also provided a lower bound construction, establishing that this rate $p = O(n^{1/2})$ cannot be improved. We strengthen this lower bound by proving that, for any constant $k$, this range of $p$ is still unimprovable even if the spanner is allowed $+k$ additive error among the demand pairs. This negatively resolves an open question asked by Coppersmith and Elkin [SODA ’05] and again by Cygan, Grandoni, and Kavitha [STACS ’13] and Abboud and Bodwin [SODA ’16].

At a technical level, our lower bounds are obtained by an improvement to the entire obstacle product framework used to compose “inner” and “outer” graphs into lower bound instances. In particular, we develop a new strategy for analysis that allows us to compose “inner” and “outer” graphs into lower bound instances.

Index Terms—Graph Theory, Lower Bounds, Spanners

I. INTRODUCTION

A basic question arising in robotics [1], circuit design [2–4], distributed algorithms [5, 6], and many other areas of computer science (see survey [7]) is to compress a graph metric $G$ into small space while approximately preserving its shortest path distances. When this compression is achieved by a sparse subgraph $H \subseteq G$ whose distance metric is similar to that of $G$, we call $H$ a spanner of $G$. The setting of spanners on a linear or near-linear number of edges is considered particularly important in applications; that is, for an input graph on $n$ nodes, we often want spanners on $O(n)$ or perhaps $O(n^{1+\epsilon})$ edges [7].

Spanners were first abstracted by Peleg and Upfal [8] and Peleg and Ullman [9] after arising implicitly in prior work. Their initial work studied spanners in the setting of multiplicative error:

Definition 1 (Multiplicative Spanners). For a graph $G$, a subgraph $H \subseteq G$ over the same vertex set is $a\cdot k$ multiplicative spanner of $G$ if, for all nodes $s, t$, we have $\text{dist}_H(s, t) \leq k \cdot \text{dist}_G(s, t)$.

Optimal bounds for multiplicative spanners were quickly closed in a classic paper by Althöfer, Das, Dobkin, Joseph, and Soares [10]. For the specific case of $O(n)$-size spanners, they proved:

Theorem 1 ( [10]). Every $n$-node graph has a $\cdot O(\log n)$ multiplicative spanner on $O(n)$ edges. Moreover, there are graphs that do not have an $\cdot O(\log n)$ multiplicative spanner on $O(n)$ edges.

Thus, the question of compression by multiplicative spanners was closed. However, for many problems, the paradigm of multiplicative error is considered unacceptable. For example, $\cdot O(\log n)$ multiplicative blowup in travel time would be unacceptable for a cross-country trucking route. This generated interest in the community in new error paradigms that perform better on the long distances in the input graph. Several new types of spanners were suggested and studied in the following years [11, 12]. The most optimistic was purely additive spanners, where pairwise distances in the spanner increase only by an additive error term:

Definition 2 (Additive Spanners [13]). For a graph $G$, a subgraph $H \subseteq G$ over the same vertex set is $+k$ additive spanner of $G$ if, for all nodes $s, t$, we have $\text{dist}_H(s, t) \leq \text{dist}_G(s, t) + k$.

Unfortunately, high-quality constructions of near-linear-size spanners with purely additive error remained elusive. That is, the community faced the following question:

Do all graphs admit $+k$ additive spanners of near-linear size, with $k$ a constant, or at least a small enough function of $n$ to be practically efficient?

This problem became a central focus of the community following a sequence of upper bound results. First was the seminal 1995 work of Aingworth, Chekuri, Indyk, and Mot-

The full version of this work can be found at arXiv:2207.11832. This work was supported by NSF-AF 2153680.
wani [14], which proved that every \( n \)-node graph has a \( +2 \) additive spanner on \( O(n^{3/2}) \) edges. This was followed by a theorem of Chechik [15] that all graphs have \( +4 \) spanners on \( O(n^{1/3}) \) edges (see also [16]), and from Baswana, Kavitha, Mehlhorn, and Pettie [17] who proved that all graphs have \( +6 \) spanners on \( O(n^{1/3}) \) edges (see also [18], [19]). Thus, it seemed that one might be able to continue this trend, paying more and more constant additive error in exchange for sparser and sparser spanners. Unfortunately, a barrier to further progress was discovered by Abboud and Bodwin [20], who constructed graphs for which any additive spanner on \( O(n^{1/3-c}) \) edges suffers \( +n^{O(1)} \) additive error. Thus, near-linear spanners with subpolynomial error are not generally possible.

That said, the lower bound of [20] is a small enough polynomial to be entirely practical even on enormous input graphs. This work showed graphs for which any \( O(n) \)-size spanner pays at least \( +n^{1/22} \) error, which is a small enough polynomial to be entirely practical even on huge input graphs.

However, the upper bounds for \( O(n) \)-size spanners were far from this ideal. The first nontrivial constructions of \( O(n) \)-size spanners [21] (following [22]) had \( +n^{9/16} \) error. Thus began a focused effort by the community to narrow these upper and lower bounds on additive error towards the middle, with the goal to determine whether practically-significant constructions of near-linear additive spanners can be generally achieved.

Despite a high throughput of recent work on the topic, the attainable error bounds for linear-size additive spanners remain wide open. The initial lower bound of \( +n^{1/22} \) additive error was improved to \( +n^{1/11} \) in two concurrent papers [23], [24], and then recently to \( +n^{1/10.5} \) by Lu, Vassilevska Williams, Wein, and Xu [25] where it stands today. Meanwhile, the upper bound on additive error was improved to \( +O(n^{1/2}) \) [26] and then to \( +n^{3/7+\varepsilon} \) [27] (see also [15], [21]).

A. New Lower Bounds

The main results of this paper are two new lower bound for additive spanners:

**Theorem 2 (First Main Result).** There are \( n \)-node graphs that do not admit a \( +O(n^{1/7}) \) additive spanner on \( O(n) \) edges.

Our construction follows the obstacle product framework used to prove lower bounds in prior work, but with a key generalization. Previously, the obstacle product carefully composes a layered outer graph with a set of layered inner graphs in a way that causes the shortest paths in the composed graphs to inherit desirable structural properties from each. We provide a stronger framework for analysis that allows one to compose unlayered inner and outer graphs.

It turns out that this unlayering technique also addresses a related open problem in the area. One can relax additive spanners to pairwise additive spanners, where we only need to approximate distances among a set of demand pairs \( P \) taken on input:

**Definition 3 (Pairwise Additive Spanners [28], [29]).** For a graph \( G = (V, E) \) and a set of demand pairs \( P \subseteq V \times V \), a subgraph \( H \subseteq G \) over the same vertex set is a \( +k \) additive spanner of \( G \) if, for all nodes \( (s, t) \in P \), we have \( \text{dist}_H(s,t) \leq \text{dist}_G(s,t) + k \).

We can then hope for better error bounds in the setting where the number of demand pairs \( |P| \) is not too large. Pairwise spanners were introduced by Coppersmith and Elkin [28], who specifically focused on the exact \( k = 0 \) case (also called distance preservers); the approximate \( k > 0 \) case has been studied repeatedly in followup work [19], [29]–[33]. The initial paper by Coppersmith and Elkin [28] established the following fundamental result:

**Theorem 3 (28).**

- **(Upper Bound)** For any \( n \)-node graph \( G = (V, E) \) and set \( P \subseteq V \times V \) of \( |P| = O(\sqrt{n}) \) demand pairs, there is a distance preserver (\(+0\) pairwise spanner) on \( O(n) \) edges.

- **(Lower Bound)** For any \( p = \omega(\sqrt{n}) \), there are \( n \)-node graphs \( G = (V, E) \) and sets of \( |P| = p \) demand pairs that do not admit a distance preserver on \( O(n) \) edges.

Thus, with a budget of \( O(n) \) edges, we can exactly preserve distances among \( O(\sqrt{n}) \) demand pairs and no more. A question asked repeatedly in followup work [28]–[30] is whether this \( \sqrt{n} \) threshold can be improved if we allow constant \( +k \) error; this question was explicitly studied in [29], [30], without resolution. We settle this question negatively:

**Theorem 4 (Second Main Result).** For any constant \( k > 0 \) and \( p = \omega(\sqrt{n}) \), there are \( n \)-node graphs \( G = (V, E) \) and sets of \( |P| = p \) demand pairs that do not admit a \(+k\) pairwise spanner on \( O(n) \) edges.

On a technical level, this stronger lower bound is again proved using an unlayered obstacle product. In our view, our new lower bound further cements the importance of the \( \sqrt{n} \) demand pair \( O(n) \) size threshold by Coppersmith and Elkin: it is even robust to any constant additive error.

II. TECHNICAL OVERVIEW OF MAIN RESULT AND COMPARISON TO PRIOR WORK

At a technical level, our main result departs from prior work by altering a fundamentally different piece of the construction than the one typically considered previously. We overview the construction and our improvement in the next section. Here we overview the parts of our construction that match prior work, and we overview the new technical ingredients that give our improved lower bounds.

A. The Obstacle Product Framework

Like every other lower bound in the area, we follow the obstacle product framework. This framework involves composing an outer graph and many copies of an inner graph.

The essential property of the outer graph \( G_O \) is that it has a set of critical pairs \( P_O \subseteq V(G_O) \times V(G_O) \), such that:

- for each critical pair \((s, t)\) there is a unique shortest \( s \leadsto t \) path \( \pi(s, t) \) in \( G_O \), called the canonical path, and
• the canonical paths are pairwise edge-disjoint.

If we remove an edge from a canonical path \( \pi(s, t) \), then since it is a unique shortest path, \( \text{dist}(s, t) \) will increase by at least \(+1\). To amplify this error, we apply an edge-extension step in which we add \( k = n^{O(1)} \) new nodes along every edge. Thus, removing an edge from an (edge-extended) canonical path \( \pi(s, t) \) will cause \( \text{dist}(s, t) \) to increase by at least \(+k\).

![Fig. 1. Deleting an edge from a canonical path stretches its distance by at least \(+1\). After the edge extension step, deleting an edge stretches distance by at least \(+k\).](image)

Although edge-extension forces any nontrivial spanner of the outer graph to suffer \(+k\) distance error, the problem is that the extended outer graph is now very sparse: the vast majority of the nodes are new edge-extension nodes of degree 2, while only a small handful of nodes are from the original outer graph and may have higher degree. Thus the trivial spanner, that keeps the entire outer graph, has \( O(n) \) size (where \( n \) is the number of nodes after edge extension) and can be used. The purpose of the next inner graph replacement step is to regain this lost density, so that an \( O(n) \)-size spanner actually has to remove a significant number of edges from the graph.

The inner graph \( G_I \) is equipped with a set of critical pairs \( P_I \) with unique edge-disjoint shortest paths, just like the outer graph. We will call these inner-canonical paths, and the paths in the outer graph outer-canonical paths to make clear the distinction. Let \( v \) be a node in the outer graph that is contained in exactly \( d \) outer-canonical paths, and suppose the inner graph \( G_I \) has exactly \( |P_I| = d \) critical pairs (the case where these quantities only approximately match, instead of both being exactly \( d \), can be handled easily). We then replace the node \( v \) with a copy of the entire graph \( G_I \). To preserve outer-canonical paths, we associate each outer-canonical path \( \pi(s, t) \) that intersects \( v \) to some inner-canonical path \( \pi(s_I, t_I) \) in \( G_I \). We arrange the composed graph in such a way that the unique shortest \( s \sim t \) path in the composed graph is exactly the original outer-canonical path \( \pi(s, t) \), with each node \( v \) replaced by the corresponding inner-canonical path \( \pi(s_I, t_I) \) in the copy of \( G_I \) that replaced \( v \). We call this unique shortest \( s \sim t \) path the composed-canonical path.

![Fig. 2. To regain lost density from edge extension, the inner graph replacement step replaces nodes in the outer graph with copies of the inner graph, carefully attaching canonical paths in the outer graph to canonical paths in the inner graph.](image)

Most of the edges in the composed graph lie in inner graphs. This implies that, in any \( O(n) \)-size spanner \( H \) of the composed graph \( G_O \otimes G_I \), there will be a composed-canonical path \( \pi(s, t) \) where the spanner is missing most of the edges used by \( \pi(s, t) \) in inner graphs. We use two cases to argue that \( \text{dist}_H(s, t) \) must be much longer than \( \text{dist}_G(s, t) \). In the first case, suppose that the shortest \( s \sim t \) path \( \pi(s, t) \) in the spanner avoids these gutted inner graphs by instead following a path that corresponds to a non-canonical \( s \sim t \) path in the outer graph. This other kind of path also suffers \( +\Theta(k) \) error over the composed-canonical path, essentially due to the edge-extension step.

![Fig. 3. In any \( O(n) \)-size spanner \( H \), there is a canonical path \( \pi(s, t) \) that is missing most of its edges in inner graphs. The additive error of \( s \sim t \) paths in the spanner are analyzed in two types: those that suffer \(+1\) error in each inner graph (left), and those that avoid the problematic inner graphs entirely by following a non-outer-canonical path (right).](image)

**B. Improvements in Prior Work: Changes to the Alternation Product**

Essentially every major improvement in the lower bound has thus far been achieved by an improvement to the alternation product. Although the alternation product is not used at all in the technical part of this paper, it is worth overviewing, to highlight the core difference between the new improvements in this paper and those obtained in prior work.
The motivating observation behind the alternation product is that, for correctness of the lower bound, one needs that each edge in an inner graph is only used by one composed-canonical path. To enforce this, it is actually overkill to require edge-disjoint outer-canonical paths. Rather, we can allow the pairwise intersections of outer-canonical paths to contain at most one edge.

Fig. 4. We can allow the outer-canonical paths to contain an edge in their pairwise intersections (left), and then after the inner graph replacement step, the composed-canonical paths will become edge-disjoint on inner graphs (right).

Thus, there is some additional freedom in outer graph design. The alternation product tries to leverage this freedom for a stronger lower bound. It changes the edge-disjoint outer-canonical paths into 2-path-disjoint outer-canonical paths, in exchange for different relative counts of nodes, critical pairs, and canonical path lengths in the outer graph. These parameter changes are favorable when the goal is to build a lower bound against denser spanners; for example, the alternation product is necessary to establish the existence of graphs that need \( n^{\Omega(1)} \) error for any spanner on \( O(n^{4/3 - \varepsilon}) \) edges. But, in a naive implementation of the alternation product, these parameter changes are unfavorable when the goal is lower bounds against \( O(n) \)-size spanners.

The improved lower bounds of \( \Omega(n^{1/11}) \) from [23], [24] are mostly achieved by removing the alternation product from [20] entirely. The subsequent \( \Omega(n^{1/10.5}) \) lower bounds of Lu et al [25] reintroduce the alternation product; their main technical innovation is a clever new implementation of the alternation product that takes advantage of the geometric structure of the outer graph to obtain more favorable parameter changes, which make it beneficial even in the setting of sparse spanners.

C. Improvements in Our Work: New Inner/Outer Graphs

The current paper obtains achievements in a different way from prior work. We omit the alternation product entirely; in that sense, our construction forks [23], [24] rather than [25] (we briefly explain why in the following section). Rather, we enable a significant technical change in the design of inner/outer graphs, which we explain next.

Roughly, the goal of the inner/outer graphs is to pack in as many critical pairs as possible, with as long canonical paths as possible. The main technical innovation in the \( \Omega(n^{1/22}) \) lower bound of [20] was to replace the “butterfly” outer graph implicitly used by Woodruff [34] with a “distance preserver lower bound graph,” along the lines of a construction by Coppersmith and Elkin [28] (see also [35], [36] for prior graph constructions based on a similar technique). Ideally, one would like to use the Coppersmith-Elkin distance preserver lower bound graphs exactly for inner/outer graphs. But there’s a catch. When we execute the inner graph replacement step, we need to make sure that the composed-canonical paths are unique shortest paths in the composed graph. This property is not immediate, and in fact it does not hold for arbitrary choices of inner/outer graphs. Rather, it holds if the inner and outer graphs are both layered. But layeredness is not free; introducing layeredness to the Coppersmith-Elkin construction significantly harms the inner/outer graph quality, leading to worse lower bounds. All previous lower bound constructions [20], [23]–[25], [34] pay this penalty in order to layer their graphs.

The technical contribution of our paper can be summarized as follows:

**Lemma 1** (Main Technical Lemma, Informal). There are certain amended versions of the Coppersmith-Elkin graphs in [28] whose structure allows them to be used as inner/outer graphs in the obstacle product, despite being unlayered.

In addition to its use in spanner lower bounds, this technical lemma is also the essential missing ingredient towards our extension of pairwise distance preserver lower bounds of Coppersmith and Elkin to pairwise additive spanners with \(+k\) error (Theorem 4). Our lower bound matches the \( \sqrt{n} \) demand pair threshold obtained by the Coppersmith-Elkin lower bound against distance preservers precisely because we can use an amended Coppersmith-Elkin distance preserver lower bound for the outer graph of our obstacle product.

One can view the Coppersmith-Elkin graph construction as parametrized by a convex set of vectors taken on input. The original graphs in [28] use a standard convex set construction from [37]. We need to design very precise convex sets that have essentially the same size as those used by Coppersmith and Elkin, but with some additional technical properties that enable use with the obstacle product. Some of our main new technical contributions lie in the design of these convex sets, which we describe in the full version of this work.

The other major technical step in this paper lies in the part of the obstacle product where outer-canonical paths are attached to inner-canonical paths in the inner graph replacement step. In all prior work, it has been completely arbitrary which outer-canonical path was attached to which inner-canonical path. In this work, it is non-arbitrary: we show that the obstacle product benefits from a specific alignment between these paths; roughly, outer-canonical paths are attached to inner-canonical paths based on the direction they are travelling. Details are given in the full version of this work.

With this, we employ a more complex version of the error analysis from prior work, that leverages our convex set design and alignment between inner and outer canonical paths. We introduce a move decomposition framework to do so, which enables an amortized version of the convexity argument used for error analysis in prior work. A high-level overview of the move decomposition framework can be found in Section IV, but we will not overview it further here.
D. Future Directions: Can the Obstacle Product Achieve Tight Error Bounds?

This research project was initiated by a thought experiment: is it even conceivable for the obstacle product framework to produce lower bounds matching the upper bounds obtained by the path-buying framework currently used for the spanner upper bounds in this paper and in [17], [27]? In this subsection, we argue that the answer is a resounding “sort of:” on all axes except one, there is clear potential for these frameworks to produce matching upper and lower bounds. This problematic axis likely spoils the possibility of pinning down an exact error bound for $O(n)$-size spanners in the near future, but this problematic axis is more of a barrier in analysis than in construction.

Suppose we run the current state-of-the-art upper bound construction from [27] on lower bound graphs produced by the obstacle product. One arrives at three main points of technical disagreement, where the upper bound analysis makes pessimistic assumptions not actually realized on the current lower bound graph. Thus, a hypothetical tight analysis would have to either introduce a lower bound graph that realizes these pessimistic assumptions, or it would need to improve the upper bound argument to avoid making these pessimistic assumptions in the first place. We overview these points, and their implications for future work, below.

a) Our Unlayered Inner Graphs Are Probably Necessary: The path-buying framework used in previous spanner upper bounds [17], [19], [38] begins with a clustering phase, in which the graph nodes are partitioned into low-radius “clusters.” In the second path-buying phase, one adds a collection of shortest paths that connect far-away clusters with small additive error. A key piece of the upper bound analysis argues that we only add a small number of shortest paths through each cluster. More specifically, leveraging distance preserver upper bounds from [28] for a cluster $C$ with $n_C$ nodes, we can afford $O(n_C^{1/2})$ shortest paths while paying only a linear number of edges for this cluster $C$.

When the path-buying framework is run on an obstacle product construction, it precisely picks out the inner graphs (plus a few nodes in the attached edge-extended paths) as clusters, and it picks out the outer-canonical paths as the shortest paths added in the second phase. Thus, for tightness, one would hope that parameters balance in such a way that we have $\Omega(n_I^{1/2})$ canonical paths through each inner graph on $n_I$ nodes. Prior to this work, this was not so: the forced layered structure of the inner graphs meant that one actually had to place $\Omega(n_I^{2/3})$ canonical paths [23] or even $\Omega(n_I)$ canonical paths [20], [25] to achieve a lower bound. By unlayering our inner graphs, we are able to rebalance parameters to have $\Omega(n_I^{1/2})$ canonical paths through each inner graph for the first time. Thus, this particular axis is no longer a point of disagreement following our work; we view our main conceptual contribution as demonstrating tightness between the path buying and obstacle product frameworks in this regard.

b) An Improved Alternation Product Is Probably Still Needed: In the clustering phase of the path-buying framework, each cluster is either “small” or “large,” depending on its number of nodes. The worst-case input graphs for the spanner upper bounds would have the structure that all clusters are right on the small/large borderline; it is a good case when all clusters are significantly above or below this threshold. As mentioned, when one clusters obstacle product graphs, the clusters are precisely the inner graphs plus some of their attached edge-extended paths. However, they would specifically be classified as large clusters in the upper bound, far away from this borderline. So the upper bound makes a pessimistic assumption of borderline clusters that is not actually realized in the current lower bound construction.

Let us engage for a moment in some wishful thinking. Suppose that we could apply an alteration product on the outer graph, and then replace in inner graphs with our current density of canonical paths but with the additional structure $S \times S$ on their demand pairs (such graphs are constructed in [28]). This would substantially reduce the number of edge-extended paths attached to each inner graph, and this change would put our inner graphs right on the small/large borderline when viewed as clusters. Thus, to resolve this discrepancy between the path buying and obstacle product frameworks, we think that the alternation product or something similar is very likely to be the right tool.

There is no intrinsic barrier to applying an alternation product on top of our unlayering method, but it complicates the already-delicate geometric details of our argument in a way that we have not resolved. So we wish to emphasize that the improved alternation product in [25] remains an important technical idea, and the natural next step for the area is to integrate this alternation product (or perhaps one with even further-improved parameters) with our unlayered inner graphs. In this sense, although our lower bounds improve numerically on [25], we think it is more conceptually correct to consider our respective constructions as concurrent state-of-the-art that achieve two different desirable features of the ultimate lower bound construction, which will need to be unified in future work.

c) Optimal Outer Graphs Will Probably Be Hard To Achieve: In order to replace in inner graphs of nontrivial size, we need to start the obstacle product construction with a relatively dense outer graph, that has poly($n$) canonical paths passing through a typical node. Such an outer graph would essentially need to be a distance preserver lower bound as in [28] (perhaps passed through an obstacle product). The trouble is that it is arguably out of reach of current techniques to determine the optimal quality of a distance preserver lower bound graph. Distance preserver lower bounds have close relationships to several long-standing open problems in extremal combinatorics, like bounds for the triangle removal lemma [39], and it will probably be difficult to settle the bounds for distance preservers without also making a breakthrough on these difficult combinatorial problems.

This paper is the first that is able to use state-of-the-art dis-
III. CONSTRUCTION FRAMEWORK

We now present our lower bound construction framework. We refer back to the technical overview (Section II) for intuition, comparison to prior work, and to highlight the part of this paper that is new. For simplicity of presentation, we will frequently ignore issues related to non-integrality of expressions that arise in our analysis; these issues affect our bounds only by lower-order terms. We direct readers looking for the complete, explicit construction of our lower bound graph to the full version of this work ([40]).

A. Base Graph $G_B$

We start by describing a template for a base graph $G_B$, which is a generalized version of the lower bound construction for distance preservers by Coppersmith and Elkin [28] (they use the following construction with a particular choice of $W$). The outer and inner graphs in our obstacle product will both be versions of the base graph, instantiated a bit differently. Graph $G_B$ will have parameters $(x, y, r, W)$.

a) Vertices: The vertices of the base graph are $[1, x] \times [1, y]$, where $x, y$ are positive integers that are inputs to the construction. We imagine these vertices as a subset of $\mathbb{Z}^2$, i.e., embedded as points in the Euclidean plane.

b) The Strongly Convex Set $W$: The edges and critical pairs of the base graph both depend on an additional input $W$, which is required to be a strongly convex set of vectors in $\mathbb{Z}^2$. We recall the definition:

Definition 4 (Strongly Convex Set). A set of vectors $W$ is strongly convex if the equation $\bar{v}_0 = \lambda_1 \bar{v}_1 + \cdots + \lambda_k \bar{v}_k$ has no nontrivial solutions with all $\bar{v}_i \in W$, $k$ any positive integer, and $\lambda_i$ (possibly negative) scalars with $\sum |\lambda_i| \leq 1$. The trivial solutions are when $\bar{v}_0 = \bar{v}_1 = \cdots = \bar{v}_k$.

We will write our strongly convex set as $W(r, \psi)$ to mean that (1) the $x$-coordinate of all vectors is between $r/2$ and $r$, and (2) the angle between any vector and the horizontal is in the range $[0, \psi]$ radians. For a technical reason to follow, we further require that the parameter $r$ satisfies $r \leq \frac{\pi}{4}$.

c) Critical Pairs: We next define the set of critical pairs $P$ for the base graph. Let $r$ be an integer parameter of $G_B$ and $W(r, \psi)$ be our chosen strongly convex set. Let $S = [1, r/2] \times [1, y/2]$, and let $T = [x - r, x] \times [1, y]$. The critical pairs $P$ are a subset of $S \times T$. Specifically: for each $s \in S$ and each $\bar{v} \in W(r, \psi)$, let $t = s + k\bar{v}$ where $k$ is the largest positive integer such that $t \in V$, and include $(s, t) \in P$. We quickly confirm that this node $t$ is well-defined:

Lemma 2. If we choose $\psi$ so that $0 \leq \psi \leq \pi/4$ and $\tan \psi \leq yx^{-1}/2$, then for all $s \in S, \bar{v} \in W(r, \psi)$ there exists a positive integer $k$ with $t := s + k\bar{v} \in T$.

Proof. Choose $k$ to be the largest integer such that $s_1 + kv_1 \leq x$, where $s_1, v_1$ are the first coordinates of $s, \bar{v}$ respectively. Since $\|\bar{v}\| \leq r$, this implies $s_1 + kv_1 \geq x - r$, and so the first coordinate of $s + k\bar{v}$ is in the appropriate range $[x - r, x]$.

For the second coordinate: since $s_2 \geq 1$ and $\tan \psi \leq 0$, we have $s_2 + kv_2 \geq 1$. Additionally, since $s_2 \leq y/2$ and $\tan \psi \leq yx^{-1}/2$, we have $s_2 + kv_2 \leq y/2 + x \cdot yx^{-1}/2 \leq y$. Thus the second coordinate of $s + k\bar{v}$ is in the appropriate range $[1, y]$, completing the proof that $s + k\bar{v} \in T$. \qed

This lemma imposes a mild constraint on our choice of $\psi$, which will be satisfied in the instantiation of our inner and outer graphs from this base graph.

d) Edges and Canonical Paths: Each critical pair $(s, t) \in P$ is generated using a vector $\bar{v} \in W(r, \psi)$; we call this the canonical vector of $(s, t)$. We define the canonical $(s, t)$-path $\pi^{s,t}_B$ the $(s \rightarrow t)$-path containing exactly the edges of the form $(s + i\bar{v}, s + (i+1)\bar{v})$ for integers $0 \leq i < k$. The edges of the graph are exactly those contained in any canonical path.

Fig. 5. For each node $s \in S$, we use the strongly convex set $W(r, \psi)$ to generate the the nodes $t \in T$ for which $(s, t)$ is included as a critical pair, in addition to the generation of the canonical paths connecting these critical pairs.

e) Important Properties of the Base Graph: This completes the construction of the base graph; we note its important structural properties before moving on. A version of this lemma appears frequently in prior work.
Lemma 3 (Properties of Base Graph $G_B$, similar to lemmas in [23], [25], [41]). The base graph $G_B = (V, E)$ has the following properties:

1) $|V| = xy$
2) $|P| = \Theta((ry \cdot |W(r, \psi)|)$
3) The canonical paths $\pi_{B}^{s,t}$ are pairwise edge-disjoint.
4) For all $s, t \in P$, $|\pi_{B}^{s,t}| = \Theta(\frac{r}{x})$. Consequently, $|E| = \Theta(xy \cdot |W(r, \psi)|)$.
5) Each canonical path $\pi_{B}^{s,t}$ is the unique shortest $(s,t)$-path in $G_B$.

Proof.

1) The number of vertices is immediate from construction.
2) There is exactly one critical pair in $P$ for each combination of a vertex in $S$ and vector in $W(r, \psi)$. Thus

$$|P| = |S| \cdot |W(r, \psi)| = \Theta(ry \cdot |W(r, \psi)|).$$

3) Each edge $(a, b) \in E$ uniquely identifies a vector $\vec{v} \in W(r, \psi)$. Since by construction $(a, b)$ lies on a canonical path, we can subtract $\vec{v}$ from a zero or more times to reach a node in $S$; since the first coordinate of $v$ is at least $r/2$ and the width of $S$ is $r/2$, there is a unique node in $S$ that we can reach in this way. Thus $(a, b)$ also uniquely identifies the first node of its canonical path $s \in S$. Since $s, \vec{v}$ determine a canonical path, $(a, b)$ lies on a unique canonical path.

4) Let $\vec{v}$ be the canonical vector of a critical pair $(s,t)$. Notice that

$$|\pi_{B}^{s,t}| = \frac{|t-s|}{\|\vec{v}\|}.$$

Since $P \subseteq S \times T$, the first-coordinate displacement $|t_1-s_1|$ is at least $x - 2r \geq x/2$. Then since $\|\vec{v}\| \leq r$, we have

$$|\pi_{B}^{s,t}| = \frac{|t-s|}{\|\vec{v}\|} \geq \frac{x-2r}{r} \geq \frac{x}{2r},$$

where the last inequality is since we require $r \leq x/4$. Since critical paths are edge-disjoint and every edge lies on a critical path, it follows that

$$|E| = |P| \cdot \Theta\left(\frac{x}{2r}\right) = \Theta(xy \cdot |W(r, \psi)|).$$

5) Let $\pi_{B}^{s,t} = (s, s+\vec{v}, s+2\vec{v}, \ldots, t)$ be a canonical path, where $\pi_{B}^{s,t}$ has $k$ edges and so $t-s = k\vec{v}$. Suppose for the sake of contradiction that there is some other $(s,t)$-path $\pi$ of length $j \leq k$ in $G_B$, and let $\vec{v}_i$ be the vertex used to create the $i$th edge in $\pi$. Then

$$k\vec{v} = t-s = \sum_{i=1}^{j} \vec{v}_i.$$

Since we have assumed that $\pi \neq \pi_{B}^{s,t}$, this violates strong convexity property of $W(r, \psi)$, completing the contradiction.

B. Composing the Final Graph $G$

a) Inner Graph and Outer Graph: We instantiate two different copies of our base graph, which we will call the inner graph $G_1 = (V_I, E_I)$ and the outer graph $G_O = (V_O, E_O)$. We will use subscripts $I$ and $O$ to indicate the inputs used to create $G_1, G_O$ respectively. That is: the inner graph has dimensions $x_1, y_1$, strongly convex set $W_I(r_1, \psi_1)$, critical pairs $P_I$, and canonical paths $\pi_{I}^{s,t}$. The outer graph parameters are the same with subscript $O$. Since $G_1, G_O$ are both instantiations of the base graph, they both satisfy Lemma 3.

b) Inner Graph Replacement Step: The next step in the construction of $G$ is to replace each vertex in $G_O$ with a copy of the inner graph $G_1$. For each vertex $u$ in $G_O$ replaced with a copy $G^u_I$ of $G_1$, we must reconnect every edge originally incident to $u$ in $G_O$ to some vertex in $G^u_I$. Recall that the critical pair set $P_I$ is a subset of $S_I \times T_I$. We will regard the source vertices $S_I$ to be the input ports of $G^u_I$ and the sink vertices $T_I$ to be the output ports of $G^u_I$. We will attach every incoming edge of form $(u-v, u)$ in $G_O$ where $\vec{v} \in W_O$ to an input port in $G^u_I$. Likewise, we will attach every outgoing edge of form $(u, u+\vec{v})$ in $G_O$ to an output port in $G^u_I$.

To perform this attachment, we define a bijection $\phi : W_O \rightarrow P_I$ from the vectors in the outer graph’s strongly convex set to the critical pairs in $P_I$. Later in the analysis, we will specify that $\phi$ is non-arbitrary; for technical reasons we must specifically choose a bijection satisfying certain properties. But for now we will prove some useful properties of the construction that hold regardless of which bijection $\phi$ is used. We note that since $\phi$ is a bijection we gain constraints

$$|W_O| = |P_I| = \Theta(ry_{I} \cdot |W_I|).$$

If $\vec{v} \in W_O$ and $\phi(\vec{v}) = (s, t)$, we plug in the incoming edge $(u-v, u)$ into input port $s$ in $G^u_I$, and we plug in the outgoing edge of form $(u, u+\vec{v})$ originally in $G_O$ into output port $t$ in $G^u_I$. We repeat this process for all copies of $G_1$ and all vectors $\vec{v} \in W_O$. Let $G'$ be the graph resulting from this process.

c) Edge Subdivision Step: Let $z$ be a new parameter of the construction. To obtain our final graph $G$, we subdivide every edge of $G'$ corresponding to an original edge in outer graph $G_O$ into a path of length $z = |V_I| \cdot |P_I|^{-1}$, by adding new nodes along the edge. We refer to the paths in $G$ replacing edges from $G_O$ as subdivided paths.

d) Critical pairs: We define the critical pairs $P$ associated to the final graph $G$ as follows:

- For each $(s_O,t_O) \in P_O$, let $G_I^{s_O}$ and $G_O^{s_O}$ be the inner graph copies in $G$ corresponding to vertices $s_O$ and $t_O$ in $G_O$. Let $\vec{v}_O$ be the canonical vector corresponding to critical pair $(s_O, t_O) \in P_O$, and let $\phi(\vec{v}_O) = (s_I, t_I) \in P_I$.
- We then add a critical pair to $P$ from the vertex $s_I$ in $G_I^{s_O}$ paired with the vertex $s_I$ in $G_I^{t_O}$. Denote these vertices in $G$ as $s$ and $t$, respectively.

1 In principle we could use $t_I$ in place of $s_I$ in $G_I^{s_O}$, but using $s_I$ instead happens to simplify some technical details later on.
• The associated canonical \((s, t)\)-path \(\pi^{s,t}\) through the final graph \(G\) is the one obtained by starting with \(\pi^{s_0,t_0}_G\) and replacing each edge with the corresponding subdivided path in \(G\) and each node \(u\) with the canonical path \(\pi^{v_i, t_i}_G\) in the graph \(G^*_i\), except that we replace the final node \(t_0\) with the single node \(s_1\) in \(G^{s_1}_G\). We define vector \(\vec{v}_O \in W_O\) to be the canonical vector of \(G\) associated with \(\pi^{s,t}\).

We summarize the properties of our construction:

**Lemma 4** (Properties of Final Graph \(G\)). Graph \(G = (V, E)\) has the following properties:

1. \(|V| = \Theta (|V_O||V_I|)\).
2. \(|P| = \Theta (r_O yO \cdot |W_O|)\).
3. The canonical paths \(\pi^{s,t}\) for \((s, t) \in P\) are pairwise edge-disjoint.
4. For all \((s, t) \in P\), \(|\pi^{s,t}| = \Theta \left( \frac{r_O}{r_I} \cdot \frac{x_I}{r_I} \right)\). Consequently, \(|E| = \Omega (|W_I| \cdot |V|)\).

**Proof.**

1. The number of vertices in inner graph copies in \(G\) is \(|V_O||V_I|\). Now we just need to count the vertices in the subdivided paths of \(G\). Each inner graph copy \(G_I\) in \(G\) is incident to at most \(2|W_O|\) subdivided paths, each of which has length \(z = |V_I| \cdot |P_I|^{-1}\). Then since \(|P_I| = |W_O|\) by our bijection \(\phi\), the number of vertices in subdivided paths is at most \(|V_O| \cdot 2|W_O| \cdot |V_I||W_O|^{-1} = 2|V_O||V_I|\).
2. The number of demand pairs follows immediately from Lemma 3 and the fact that \(|P| = |P_O|\).
3. The fact that canonical paths do not share edges along subdivided paths follows from edge-disjointness of canonical paths in the outer graph (Lemma 3). The fact that canonical paths do not share edges in inner graph copies follows by noticing that any two canonical paths in \(G_O\) with the same canonical vector are node-disjoint. Thus, any two canonical paths in \(G\) that use the same inner graph \(G^*_i\) have different canonical vectors, and so they use different canonical subpaths through \(G^*_i\), as determined by the bijection \(\phi\). The claim then follows from edge-disjointness of canonical paths in \(G_I\).
4. Let \((s, t) \in P\), and let paths \(\pi^{s_0, t_0}\) and \(\pi^{s_1, t_1}\) be the canonical paths in \(G_O\) and \(G_I\) respectively used to define \(\pi^{s,t}\). By construction we have

\[|\pi^{s,t}| \geq |\pi^{s_0, t_0}| + |\pi^{s_1, t_1}|.\]

Applying Lemma 3 to bound the lengths of canonical paths, we thus have

\[|\pi^{s,t}| = \Theta \left( \frac{x_O}{r_O} \cdot \frac{x_I}{r_I} \right).\]

Finally, since canonical paths of \(G\) are edge-disjoint and \(|W_O| = |P_I|\), we have

\[|E| \geq |P| \cdot \Theta \left( \frac{x_O}{r_O} \cdot \frac{x_I}{r_I} \right) = \Omega \left( |W_I| \cdot |V| \right).\]

This completes the outline of the construction framework. The explicit construction of strongly convex sets \(W_I\) and \(W_O\), and bijection \(\phi\) can be found in the full version of this work ([40]).

**IV. Analysis Framework**

Fix a critical pair of vertices \((s, t) \in P\) in our final graph \(G\). Let \(\pi^*\) denote the canonical path corresponding to critical pair \((s, t)\) in \(G\), and let \(\pi\) be any alternate \(s \sim t\) path. The majority of our analysis will be dedicated to proving that \(\pi^*\) is much longer than \(\pi\) in the case where \(\pi\) takes at least one subdivided path not in \(\pi^*\); specifically, we show \(|\pi| - |\pi^*| = \Omega(r_O^2/3)\) (see Lemma 5). After proving this lemma, the rest of the analysis follows arguments similar to prior work [20], [23]–[25].

We begin our analysis by decomposing paths in \(G\) into subpaths we call moves. We define a partition of these moves that we call the movest set \(\mathcal{M}\).

**Definition 5** (Movest Set \(\mathcal{M}\)). Let \(\pi\) be a \((u, v)\)-path in \(G\) from some input port \(u \in S_I\) in some inner graph copy \(G^{(1)}_I\) to some input port \(v \in S_I\) in some inner graph copy \(G^{(2)}_I\). If no internal vertex of \(\pi\) is an input port, then we call \(\pi\) a move.

We define the following categories of moves in \(G\):

- **FORWARD MOVE.** Path \(\pi\) is a forward move if it travels from \(u\) to some output port \(w \in T_I\) in \(G^{(1)}_I\) and then takes a subdivided path \(e\) from \(w\) to reach input port \(v\) in \(G^{(2)}_I\).
- **BACKWARD MOVE.** Path \(\pi\) is a backward move if it takes some subdivided path \(e\) incident to \(u\) to reach some output port \(w \in T_I\) in \(G^{(1)}_I\) and then travels to input port \(v\) in \(G^{(2)}_I\).
- **ZIGZAG MOVE.** Path \(\pi\) is a zigzag move if it takes some subdivided path \(e_1\) incident to \(u\) to reach some output port \(w_1 \in T_I\) in some inner graph copy \(G^{(3)}_I\), then travels to some output port \(w_2 \in T_I\) in \(G^{(3)}_I\), and then takes a subdivided path \(e_2\) incident to \(w_2\) to reach vertex \(v\) in \(G^{(2)}_I\).
- **STATIONARY MOVE.** Path \(\pi\) is a stationary move if \(G^{(1)}_I = G^{(2)}_I\), i.e. if \(u\) and \(v\) are input ports in the same inner graph copy.

We define the movest set \(\mathcal{M}\) to be the collection of these categories of moves, namely

\[\mathcal{M} = \{\text{FORWARD, BACKWARD, ZIGZAG, STATIONARY}\}.\]

Moves will be the basic unit by which we analyze \((s, t)\)-paths in \(G\). A useful property of the movest is the following.

**Proposition 1.** Every simple \((s, t)\)-path \(\pi\) can be decomposed into a sequence of pairwise internally vertex-disjoint moves from the movest.

**Proof.** Let \(s_1, s_2, \ldots, s_k\) be the list of input ports contained in \(\pi\), listed in their order in \(\pi\). Note that \(s_1 = s\) and \(s_k = t\). Each subpath \(\pi[s_i, s_{i+1}]\) will have no input port as an internal vertex, and therefore will be a move \(m_i\). This move
that is, the (possibly negative) scalar projection of the vector $\vec{m}_i$ onto $\vec{v}^\pi$ in the standard Euclidean inner product.

We roughly use $d_i$ as a measure of how much closer or farther we get to $t$ in $G$ when we take move $m_i$. Besides the move distance $d_i$ of $m_i$, the other salient property is its length (number of edges) in the final graph, $|m_i|$. We will be comparing the moves of a path $\pi$ against the moves of $\pi^*$, all of which have move distance $\|\vec{v}^\pi\|$ and the same path length. The following quantity will be useful for this purpose.

**Definition 8** (Unit length of $\pi^*$). We define the unit length $L_{\pi^*}$ of $\pi^*$ as $L_{\pi^*} := \frac{\|\pi^*\|}{\|t-s\|}$.

$L_{\pi^*}$ is the number of edges in $\pi^*$ per unit distance travelled in $\mathbb{Z}^2$. Using this quantity we can directly compare any move $m_i$ to the moves of $\pi^*$ via the following quantity.

**Definition 9** (Move length difference). $\Delta(m_i) = |m_i| - L_{\pi^*} d_i$

The move length difference $\Delta(m_i)$ is the number of additional edges used by $m_i$ to travel distance $d_i$ in the direction $\vec{v}^\pi$, as compared with the same move in $\pi^*$.

**Proposition 2.** $\sum \Delta(m_i) = |\pi| - |\pi^*|$

**Proof.** We have:

$$\sum \Delta(m_i) = \sum |m_i| - L_{\pi^*} \sum d_i = |\pi| - \frac{|\pi^*|}{\|t-s\|} \sum d_i = |\pi| - |\pi^*|$$

The final equality follows from the fact that $\sum d_i = \sum \text{proj}_{\vec{v}^\pi} \vec{m}_i = \text{proj}_{\vec{v}^\pi} (t-s) = ||t-s||$, since $\pi$ is an $(s,t)$-path and $\vec{v}^\pi$ is the canonical vector of $(s,t) \in P$. \qed

Proposition 2 gives us a way to compare $|\pi|$ and $|\pi^*|$ at the level of individual moves. If we could show that for all moves $m_i$, $\Delta(m_i) \geq 0$, then we would be a lot closer to our current goal of proving a separation between $|\pi|$ and $|\pi^*|$. (Roughly speaking, the inequality $\Delta(m_i) \geq 0$ was immediate in prior constructions.) Unfortunately, this is not generally true in our construction. Because our inner graph is unlayered, it’s possible that the canonical inner graph path used by $\pi^*$ in copies of inner graph $G_I$ is much longer than a different path connecting some input port to some output port in $G_I$, which might be used in an alternate move $m_i$. This would result in negative $\Delta(m_i)$.

We will outline our fix here, although some technical details are pushed to later in the argument where they are used. We will use an amortized version of move difference, based on a potential function $\Phi : S_I \mapsto \mathbb{R}_{\geq 0}$ that we call the inner graph potential. Note that the input to $\Phi$ is an input port in the original inner graph; we will evaluate $\Phi$ on input ports in various inner graph copies in the final graph, and so if $a,b$ represent the same input port in two different inner graph copies, we must have $\Phi(a) = \Phi(b)$. We specify $\Phi$ in the full version of this work. With this, we can define an amortized version of move difference, $\Delta(m_i)$.
Definition 10 (Amortized move difference). \( \Delta(m_i) := \Delta(m_i) - (\Phi(s_{i+1}) - \Phi(s_i)) \)

The following proposition shows that the amortized move difference still captures the difference between \( |\pi| \) and \( |\pi^*| \).

Proposition 3. \( \sum_i \Delta(m_i) = |\pi| - |\pi^*| \)

Proof. We have:
\[
\sum_i \Delta(m_i) = \sum_i \Delta(m_i) - \sum_i (\Phi(s_{i+1}) - \Phi(s_i)) = |\pi| - |\pi^*| - (\Phi(t) - \Phi(s)) = |\pi| - |\pi^*|.
\]

The final equality follows from the fact that \( s \) and \( t \) have the same coordinates in their respective inner graph copies as specified in the definition of \( P \).

In the full version of the paper (see [40]) we prove that \( \Delta(m) \geq 0 \) for every move \( m \). Then using Proposition 3 and the unique shortest path property of the base graph \( G_B \) (see Lemma 3), we will obtain our desired separation between \( |\pi| \) and \( |\pi^*| \). Specifically, we prove the following lemma.

Lemma 5. Let \( \pi \) be an \((s,t)\)-path in \( G \) that takes a subdivided path not in \( \pi^* \). Then \( |\pi| - |\pi^*| = \Omega(\frac{n^{2/3}c^{-130}}{r_o^2}) \).

Proof. See the full version of this work [40].

V. FINISHING THE LOWER BOUNDS

We have established Lemma 5, our main results follow from arguments similar to those in previous works [23] [25].

Theorem 5. For any sufficiently large parameter \( c_0 \), there are infinitely many \( n \) for which there is an \( n \)-vertex graph \( G \) such that any spanner of \( G \) with at most \( c_0 n \) edges has additive distortion \( +\Omega(n^{1/2}r_o^{-80}) \).

Proof. We are given a sufficiently large parameter \( c_0 > 0 \). Then we will choose construction parameter \( c = \Theta(c_0) \) for our infinite family of graphs \( G \). As proven in the full version of this work, every graph in our family on \( n \) vertices has \( m = \Theta(cn) \) edges. We choose \( c \) to be sufficiently large so that for every graph \( G \) on \( n \) vertices and \( m \) edges in our infinite family of graphs, \( \frac{m}{n} < 1/2 \). We define our demand pairs to be the set of critical pairs \( P \) in our construction. Let \( H \) be a pairwise spanner of \((G,P)\) with at most \( c_0 n \) edges. By an argument identical to that of Theorem 5, it follows that \( H \) has additive distortion at least \( +k = \min\{\frac{n^{2/3}c^{-130}}{r_o^2}, \frac{n}{r_o}\} \).

Now instead of choosing \( r_o \) to grow polynomially with \( n \) as in Theorem 5, we will let \( r_o = \Theta(n/1) \). Moreover, we will require that \( r_o > c_0^{100} \). Note that there exists infinitely many valid choices of \( r_o \) satisfying these criteria. Then for sufficiently large \( c_0 \) and sufficiently large \( n \) relative to \( c_0 \), it follows that \( +k = \min\{\frac{n^{2/3}c^{-130}}{r_o^2}, \frac{n}{r_o}\} \geq c_0 \). Additionally, as specified in the full version of this paper, \( |P| = \Theta(r_o^{5/3}c^{-5} \cdot r_o^{1/2} \cdot n^{1/5}) = \Theta(n/1) \), so we conclude that \( |P| = \Theta(c_0^{1/2}) \).

ACKNOWLEDGMENTS

The authors would like to thank Shang-En Huang for helpful discussions about lower bound graphs.

REFERENCES

[1] L. Cai, “Np-completeness of minimum spanner problems,” Discrete Applied Mathematics, vol. 48, no. 2, pp. 187–194, 1994.
[2] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. Wong, Performance-driven global routing for cell based IC’s. University of California (Los Angeles). Computer Science Department, 1990.
[3] ——, “Provably good algorithms for performance-driven global routing,” Proc. of 5th ISCAS, pp. 2240–2243, 1992.
[4] F. S. Salman, J. Cheryian, R. Ravi, and S. Subramanian, “Approximating the single-sink link-installation problem in network design,” SIAM Journal on Optimization, vol. 11, no. 3, pp. 595–610, 2001.
[5] B. Awerbuch, “Complexity of network synchronization,” Journal of the ACM (JACM), vol. 32, no. 4, pp. 804–823, 1985.
[6] S. Bhatt, F. Chung, T. Leighton, and A. Rosenberg, “Optimal simulations of tree machines,” in 27th Annual Symposium on Foundations of Computer Science (sfcs 86). IEEE, 1986, pp. 274–282.
[7] R. Ahmed, G. Bodwin, F. D. Sahneh, K. Hamm, M. J. L. Jebelli, S. Kobourov, and R. Spence, “Graph spanners: A tutorial review,” Computer Science Review, vol. 37, p. 100253, 2020.
