Approximating the Norms of Graph Spanners

Eden Chlamtáč
Ben Gurion University of the Negev, Beersheva, Israel

Michael Dinitz
Johns Hopkins University, Baltimore, MD, USA

Thomas Robinson
Ben Gurion University of the Negev, Beersheva, Israel

Abstract

The ℓ_p-norm of the degree vector was recently introduced by [Chlamtáč, Dinitz, Robinson ICALP '19] as a new cost metric for graph spanners, as it interpolates between two traditional notions of cost (the sparsity ℓ_1 and the max degree ℓ_∞) and is well-motivated from applications. We study this from an approximation algorithms point of view, analyzing old algorithms and designing new algorithms for this new context, as well as providing hardness results. Our main results are for the ℓ_2-norm and stretch 3, where we give a tight analysis of the greedy algorithm and a new algorithm specifically tailored to this setting which gives an improved approximation ratio.

2012 ACM Subject Classification Theory of computation → Sparsification and spanners

Keywords and phrases Spanners, Approximations

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.11

Category APPROX

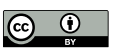
Funding *Eden Chlamtáč*: Supported in part by ISF grant 1002/14.
Michael Dinitz: Supported in part by NSF awards CCF-1464239 and CCF-1535887.
Thomas Robinson: Supported in part by ISF grant 1002/14.

1 Introduction

Graph spanners are subgraphs which approximately preserve distances: given a graph $G = (V, E)$ (possibly with lengths on the edges), a subgraph H of G is a *t-spanner* of G if $d_G(u, v) \leq d_H(u, v) \leq t \cdot d_G(u, v)$ for all $u, v \in V$, where d_G denotes shortest-path distances in G (and d_H in H). The value t is called the *stretch* of the spanner.

There have been two traditional ways of studying spanners. The first way is to study universal tradeoffs that can be achieved in all graphs between the stretch and some notion of the "cost" of a spanner, particularly the sparsity [2] or the weight [8]. The second is to study the optimization problem arising from fixing the stretch and trying to optimize the "cost" for the particular given graph. These two lines of work are highly complementary, and have proceeded in parallel. So there is now an extensive line of work on tradeoffs and approximation algorithms for sparsity (total number of edges) and, to a lesser extent, the maximum degree, which are two of the oldest and most well-studied notions of cost.

However, both of these objective functions have drawbacks. If we optimize the sparsity we might end up with a small number of very large degree nodes, which can be a problem for many applications (particularly in distributed systems where the degree is usually related to some notion of "load" on a node). On the other hand, if we try to minimize the maximum degree then we get the opposite problem. If it is unavoidable for there to be some node of large degree d, the maximum degree objective allows us to make every other vertex also of degree d, with no change in the objective function. Since the whole point of using spanners is to get a more compact representation of the graph, this is a significant issue.



In order to remedy these drawbacks, [10] recently proposed a new objective function: the ℓ_p norm of the degree vector. Given a spanner H, we can define $\|H\|_p$ to be the ℓ_p-norm of the n-dimensional vector in which the coordinate corresponding to a node v contains the degree of v in H. Then $\|H\|_1$ is just (twice) the total number of edges, and $\|H\|_\infty$ is precisely the maximum degree. Thus the ℓ_p-norm is an interpolation between these two classical objectives. Moreover, for $1 < p < \infty$, this notion of cost has the properties that we want: it encourages low-degree nodes rather than high-degree nodes, but if high-degree nodes are unavoidable it still encourages the rest of the nodes to be as low-degree as possible. These properties, of interpolating between the average and the maximum, are why the ℓ_p-norm has appeared as a popular objective for a variety of problems, ranging from clustering (the famous k-means problem [17, 19]), to scheduling [6, 5, 1], to covering [16].

The focus of [10] was on universal guarantees rather than approximation algorithms, although they made interesting and suggestive observations about approximation algorithms. In particular, they showed that for stretch 3 and the ℓ_2-norm, the greedy algorithm performs better than would be expected from its behavior in ℓ_1 and ℓ_∞ (see Section 1.1 for more discussion). In this paper we focus on approximation algorithms, particularly for the special case pointed out by [10] – stretch 3 and the ℓ_2-norm. We precisely characterize the performance of the greedy algorithm, showing that it does even better than was claimed in [10]. We then design a new algorithm which is specialized to this setting and which, when combined with the greedy algorithm, gives the best known approximation.

1.1 Background on ℓ_p-Norm Spanners

We will be concerned with the following problem.

▶ **Definition 1.** *In the* Minimum ℓ_p-Norm t-Spanner *problem we are given an (unweighted) graph $G = (V, E)$ and are asked to find the t-spanner H of G which minimizes $\|H\|_p$.*

In this paper we will focus on Minimum ℓ_2-Norm 3-Spanner, although many of our techniques can be extended to other stretch values and ℓ_p norms.

Recall the classical greedy algorithm for finding t-spanners in undirected graphs: we add edges to the spanner as long as they do not close a cycle of length at most $t + 1$. In the weighted setting, edges are sorted by non-decreasing order of weight, and added as long as they are not already t-spanned. Here, we focus only on the unweighted setting.

In [10], the authors gave the following tight universal bounds on the ℓ_2 norm of a 3-spanner:

▶ **Theorem 2** ([10])**.** *Given an n-vertex connected unweighted undirected graph G:*

1. *There exists a 3-spanner H of G with $\|H\|_2 \leq \min\{O(n), \|G\|_2\}$, and the greedy algorithm returns such a spanner.*
2. *Any 3-spanner H of G must satisfy $\|H\|_2 \geq \max\{\sqrt{n}, \tilde{\Omega}(\sqrt{\|G\|_2})\}$.*

This immediately implies the following approximation guarantees:

▶ **Corollary 3.** *Given an n-vertex unweighted graph G, the greedy algorithm gives an $O(\sqrt{n})$-approximation for* Minimum ℓ_2-Norm 3-Spanner.

▶ **Corollary 4.** *Given an n-vertex unweighted graph G, the greedy algorithm gives an $\tilde{O}(n/\sqrt{\|G\|_2})$-approximation for* Minimum ℓ_2-Norm 3-Spanner.

Corollary 3 is the strongest approximation guarantee, as a function of n, that follows from the universal bounds in Theorem 2. However, unlike in the ℓ_1 and ℓ_∞ case, the authors of [10] showed that such tight universal upper and lower bounds do not give a tight analysis of the approximation guarantee for ℓ_2. In particular, the authors showed that the greedy algorithm actually gives a slightly better $O(n^{63/128})$-approximation.

1.2 Our Results and Techniques

We begin in Section 2 by giving a new analysis of the greedy algorithm, improving on the $O(n^{63/128})$ bound from [10].

▶ **Theorem 5.** *Given an n-vertex unweighted graph G, the greedy algorithm gives an $\tilde{O}(n^{3/7})$-approximation for* Minimum ℓ_2-Norm 3-Spanner.

We also show that this analysis is tight, i.e., there are graphs in which greedy is an $\tilde{\Omega}(n^{3/7})$-approximation. Thus we resolve the question raised by [10] on the performance of the greedy algorithm for the ℓ_2-norm and stretch 3.

Interestingly, despite the fact that greedy is purely combinatorial, we analyze it via a constant-size linear program: we show that the problem of finding the worst-case approximation ratio of the greedy algorithm reduces to solving a single LP. To do this, we decompose the input graph into a small collection of nearly-biregular subgraphs. For any such subgraph, this LP has variables describing degree and size parameters in the relevant portion of the greedy spanner and an optimal spanner (and thus has constant size). The objective function in the LP captures the ratio of the upper bound on the ℓ_2-norm of a greedy spanner to an optimal spanner for any of these subgraphs. We find an optimal solution to this LP, thus giving a tight bound on the approximation ratio.

We then go beyond previously proposed algorithms to give a new algorithm which is specialized to the case of the ℓ_2-norm and stretch 3. First we rewrite the standard flow-based LP for spanners (from [12]) to have an ℓ_p-norm objective, which leaves it as a convex (rather than linear) program which is polynomial-time solvable via Ellipsoid. We then give two new rounding algorithms, one of which is essentially the algorithm used in [9] for the ℓ_∞-norm objective, but with different parameters and a different objective, and thus a different analysis. Our second new algorithm draws independent random values for every edge and vertex in the graph, and includes an edge e if these values satisfy one of three conditions relating to the solution of the convex relaxation. Similar ideas have been used for stretch 3 and 4 with the ℓ_1-objective [12, 7, 13], but this is the first algorithm (to the best of our knowledge) which combines vertex and edge random sampling.

While it is common to trade off two different algorithms at the parameter setting where they have the same approximation ratio (e.g., as was done in the ℓ_1-objective for spanners [12, 7]), this is not what we do. Instead, the most important question is *correctness*: we carefully parameterize these algorithms so that every edge is spanned in the combined algorithm. Proving that this combined algorithm does yield a 3-spanner, and analyzing its approximation ratio, is surprisingly complex and takes up the bulk of this paper. In the end, we prove the following theorem.

▶ **Theorem 6.** *There is a polynomial-time algorithm for* Minimum ℓ_2-Norm 3-Spanner *with approximation ratio $\tilde{O}(\|G\|_2^{5/16})$.*

Finally, trading our new algorithm off with the greedy guarantee of Corollary 4, we immediately get our strongest approximation guarantee as a function of n:

▶ **Corollary 7.** *Trying both the algorithm of Theorem 6 and the greedy algorithm and returning the better of the two gives an $\tilde{O}(n^{5/13})$-approximation.*

In light of all of these upper bound results, a natural question is whether Minimum ℓ_2-Norm 3-Spanner is also hard to approximate. This is also important because strong hardness results are known for both the ℓ_1 and ℓ_∞ norms. Strong hardness of approximation for the ℓ_1-norm in directed graphs has been known since [18, 15] (where strong means

the same hardness that is known for the famous Label Cover problem, i.e., hard to approximate better than $2^{\log^{1-\epsilon} n}$ for arbitrarily small constant ϵ), and this was recently extended to undirected graphs by [11] by proving hardness for instances of Label Cover with some extra structure. For the ℓ_∞-norm objective, the techniques of [18, 15, 11] were significantly extended in both the directed and undirected settings by [9] in order to prove similar hardness bounds.

It is not hard to see that if we attempt to use the hardness results for ℓ_1 or ℓ_∞ as a "black box" then we will not be able to prove anything useful, simply because the hardness results are subpolynomial (with respect to n) and thus changing the norm loses the entire hardness. In fact, the hardness reduction used in the ℓ_1 case [11] does not seem to work for the ℓ_2-norm, since it relies on adding many low-degree nodes to amplify the hardness. On the other hand, we show that if we use the ℓ_∞ hardness reduction of [9] (with slightly different parameters), which amplifies hardness by adding a small number of high-degree nodes, we can prove a similar hardness bound.

▶ **Theorem 8.** *Unless* NP $\subseteq$ BPTIME$(2^{\text{polylog}(n)})$*, for any constant* $\epsilon > 0$ *there is no polynomial-time algorithm that can approximate* Minimum ℓ_2-Norm 3-Spanner *better than* $2^{\log^{1-\epsilon} n}$.

At a very high level, this is obtained by re-analyzing the reduction of [9] more carefully. In [9], since they cared only about the maximum degree, it was not necessary to analyze the (many) nodes with smaller degrees. Moreover, some of the key arguments in [9] are false in the context of the ℓ_2-norm: there is an argument that we can change the optimal solution to be "canonical" without affecting the ℓ_∞ norm, but in the ℓ_2-norm there is an effect. So we need to instantiate the reduction with different parameters, perform a more detailed analysis, and replace some key steps with more refined arguments. This all significantly complicates the analysis. Since our main focus is on algorithms rather than hardness, we defer the proof to Appendix A.

2 Greedy

Here, we improve over the the analysis in [10] and give a nearly tight (up to polylogarithimic factors) analysis of the greedy algorithm. We give only a high-level overview here, and defer many of the details to the full version of the paper. We show the following:

▶ **Theorem 9.** *Given an* n*-vertex unweighted graph* G*, the greedy algorithm gives an* $\tilde{O}(n^{3/7})$*-approximation for the minimum* ℓ_2*-norm 3-spanner.*

This can be seen to be tight by considering the following graph: Let T be a tree of depth 3, where the root has $n^{4/7}$ children, all level 1 nodes have $n^{2/7}$ children, and all level 2 nodes have $n^{1/7}$ children (so the number of leaves is n). Now let G be the graph created by taking T and adding an edge between the root and every leaf. Clearly, T is a 3-spanner of G with $\|T\|_2 = O(n^{4/7})$. However, the greedy algorithm could start by taking all the edges from the root to the leaves of T, right away creating a subgraph with ℓ_2-norm at least n.

Our analysis will decompose the graph into a small number of well structured subgraphs, and analyze the behavior of the greedy algorithm on each part. The condition on each subgraph is the following.

▶ **Definition 10.** *We say a graph* (L, R, E) *with vertex set* $L \cup R$ *is* nearly bi-regular *if there exist integers* d_L, d_R *such that every vertex* $u \in L$ *has* $|\Gamma(u) \cap R| \in [d_L/6, d_L]$ *and every vertex* $v \in R$ *has* $|\Gamma(v) \cap L| \in [d_R/(6 \log |R|), d_R]$.

We use standard regularization techniques to give the following decomposition (the proof is deferred to the full version).

▶ **Lemma 11.** *Given an undirected graph $G = (V, E)$ with $|V| = n$, there exist $O(\log^3 n)$ subgraphs $H_i = (L_i, R_i, E_i)$ of G such that the edge sets $\{E_i\}$ are a partition of E, and each H_i is nearly bi-regular.*

To analyze the performance of the greedy algorithm on each subgraph in the above partition, we use a specific constant size linear program, similar to the linear program used for the universal lower bound in [10], but with a different objective function: finding the worst-case ratio between the ℓ_2-norm of the greedy algorithm and an optimal spanner. The linear program assumes that an optimal set of paths of length ≤ 3 that span the edges of any biregular graph H_i has a fairly regular structure. In particular, it assumes that the union of such an extremal set of paths is a four layered graph such that the subgraph induced on every two subsequent layers is bipartite and biregular. Such a graph can be succinctly described by the cardinalities of the different layers and the degrees of the bipartite graphs connecting every two consecutive layers. A pruning argument shows that this assumption is without loss of generality, up to a polylogarithmic factor in the ℓ_2 norm.

We solve this linear program and show that the example graph described after Theorem 5 gives a feasible solution with value $n^{3/7}$, for which there is a dual solution giving the complementary bound. This linear program, its optimal solution, and its connection to the performance of the greedy algorithm are all given in the full version. Here we only mention the conclusion:

▶ **Lemma 12.** *Let H be an N-vertex nearly bi-regular graph, and let P be a graph (not necessarily a subgraph) which spans every edge in E by a path of length at most 3. Then we have $\min\{N, \|H\|_2\}/\|P\|_2 = \tilde{O}(N^{3/7})$.*

We can now prove our main theorem for this section.

Proof of Theorem 9. Let $\{H_i\}$ be the partition of G into $O(\log^3 n)$ subgraphs given in Lemma 11, and let N_i be the number of vertices in H_i. If H is a spanner returned by the greedy algorithm, we know by Theorem 2 that for each i, we have $\|H \cap H_i\|_2 = \min\{O(N_i), \|H_i\|_2\}$. Choose an i_0 that maximizes this expression. Then we have $\|H\|_2 \leq \sum_i \|H \cap H_i\|_2 = O(\log^3 n) \cdot \min\{N_{i_0}, \|H_{i_0}\|_2\}$.

On the other hand, letting P be an optimal 3-spanner of G, we know in particular that P spans the edges in H_{i_0}. And so our approximation ratio is bounded by

$$\begin{aligned}\frac{\|H\|_2}{\|P\|_2} &\leq \frac{O(\log^3 n)\min\{N_{i_0}, \|H_{i_0}\|_2\}}{\|P\|_2} \\ &= \tilde{O}(N_{i_0}^{3/7}) && \text{by Lemma 12} \\ &= \tilde{O}(n^{3/7}).\end{aligned}$$

◀

3 LP-Based Rounding

We now turn to algorithms based on rounding LP relaxations. In particular, we analyze the performance of the linear programming relaxation (though with a different objective function) suggested by [12, 9] for Minimum ℓ_1 3-Spanner and Minimum ℓ_∞ 3-Spanner, respectively. Focusing on ℓ_2, we consider the following convex program, noting that it is only the objective function which is nonlinear:

$$\min \quad \left(\sum_{v \in V} \left(\sum_{e \sim v} x_e \right)^2 \right)^{1/2}$$

$$\text{s.t.} \quad \sum_{p: u \rightsquigarrow v, |p| \leq 3} y_p = 1 \qquad \forall (u,v) \in E \tag{1}$$

$$x_e \geq \sum_{\substack{p: u \rightsquigarrow v, |p| \leq 3 \\ p \ni e}} y_p \qquad \forall (u,v), e \in E \tag{2}$$

$$x_e, y_p \geq 0 \qquad \forall e, p \tag{3}$$

While the objective function is not linear, it is convex, and so this LP can be efficiently solved by standard techniques (e.g., the Ellipsoid Method). Let us briefly see why this is a relaxation. In the intended (integral) solution, for every edge $e \in E$, x_e is an indicator for whether e appears in our spanner. Thus the objective function describes the ℓ_2 norm of our spanner. Furthermore, for every edge (u,v) we can pick a unique path p of length at most 3 between u and v in our spanner, and set $y_p = 1$, while setting $y_{p'} = 0$ for every other path p' between u and v. This is clearly a feasible solution.

3.1 Independent Edge Sampling

Given an optimum solution to the linear program, consider the following simple rounding algorithm, which slightly generalizes the rounding suggested in [9], parametrized by a constant $\alpha \in (0,1)$:

Edge-Round(α): Independently add each edge $e \in E$ to the spanner with probability x_e^α.

One part of our rounding algorithm will use this rounding for a specific value of α, though it will not necessarily return a spanner. We would like to bound the ℓ_2 norm of the subgraph returned by this algorithm. For our anlalysis of this and other rounding algorithms, we will need the following standard Chernoff bound (cf. [14], Theorem 1.1):

▶ **Theorem 13.** *Let $X = \sum_{i=1}^n X_i$, where X_i are independently distributed in [0,1]. Then for all $t > 2e\mathbb{E}[X]$, we have* $\text{Prob}[X > t] \leq 2^{-t}$.

We have the following bound on the ℓ_p-norm of the subgraph returned by Algorithm Edge-Round:

▶ **Lemma 14.** *Let H be the output of Edge-Round(α). Then with probability at least $1 - 2^{\log n - \log^2 n}$, we have $\|H\|_2 \leq \log^2(n) \|G\|_2^{1-\alpha} \text{LP}^\alpha$.*

Proof. First note that for every vertex $v \in V$ the expected degree of v in H is

$$\mathbb{E}\left[d_H(v)\right] = \sum_{e \sim v} x_e^\alpha \geq 1,$$

where the inequality follows since the x_e's support a flow of 1 from v to any neighbor. Thus, by Theorem 13, and taking a union bound over all vertices, we have that with probability at least $1 - n2^{-\log^2 n}$ all vertices $v \in V$ satisfy

$$\begin{aligned} \mathbb{E}\left[d_H(v)\right] &\leq \log^2 n \cdot \sum_{e \sim v} x_e^\alpha \\ &\leq \log^2 n \cdot d_G(v)^{1-\alpha} \left(\sum_{e \sim v} x_e \right)^\alpha. \qquad \text{(by Hölder's inequality)} \end{aligned}$$

Thus if we define vectors f, g as $f_v = d_G(v)^{2(1-\alpha)}$ and $g_v = (d_{\text{LP}}(v))^{2\alpha}$, then we get

$$\begin{aligned}
\|H\|_2^2 &\leq \log^4 n \sum_{v \in V} d_G(v)^{2(1-\alpha)} (d_{\text{LP}})^{2\alpha} \\
&= \log^4 n \; f \cdot g \\
&\leq \log^4 n \; \|f\|_{1/(1-\alpha)} \|g\|_{1/\alpha} && \text{(by Hölder's inequality)} \\
&= \log^4 n \left(\sum_{v \in V} d_G(v)^2 \right)^{1-\alpha} \left(\sum_{v \in V} (d_{\text{LP}}(v))^2 \right)^{\alpha} \\
&= \log^4 n \|G\|_2^{2(1-\alpha)} \text{LP}^{2\alpha}
\end{aligned}$$

◀

The above lemma on its own does not give a clear approximation guarantee. However, when combined with the known lower bounds on OPT, we can get the following bound:[1]

▶ **Lemma 15.** *Let H be the output of Edge-Round(α). Then with probability at least $1 - 2^{\log n - \log^2 n}$, we have $\|H\|_2 = \tilde{O}\left(\|G\|_2^{(1-\alpha)/2}\right) \cdot \text{OPT}$.*

Proof. With the stated probability, by Lemma 14 we have

$$\begin{aligned}
\|H\|_2 &\leq \log^2 n \|G\|_2^{1-\alpha} \text{LP}^{\alpha} \leq \log^2 n \|G\|_2^{1-\alpha} \text{OPT}^{\alpha} = \log^2 n \left(\frac{\|G\|_2}{\text{OPT}} \right)^{1-\alpha} \cdot \text{OPT} \\
&\leq \log^2 n \left(\frac{\|G\|_2}{\tilde{\Omega}(\sqrt{\|G\|_2})} \right)^{1-\alpha} \cdot \text{OPT},
\end{aligned}$$

where the final inequality is from the lower bound in Theorem 2, which proves the lemma. ◀

3.2 A New Rounding Algorithm

We now present a new rounding algorithm for the same linear programming relaxation, which we have designed specifically for the ℓ_2-norm, and which gives our best approximation guarantee (when traded off with the greedy algorithm).

In fact, we will round our LP solution by trying two different algorithms, and returning the union of the edge sets returned by the two algorithms. We will show that every edge will be spanned by at least one of the two algorithms with high probability. Our first algorithm is simply Edge-Round(3/7).

Algorithm 1 Edge-Round(3/7).

Independently add each edge $e \in E$ to the spanner with probability $x_e^{3/7}$.

Lemma 15 directly implies that with high probability this algorithm returns a subgraph with ℓ_2 norm at most $\text{OPT} \cdot \tilde{O}(\|G\|_2^{2/7})$, which is even better than our final guarantee (see Lemma 17). However, it is not guaranteed to return a valid spanner.

Our second algorithm takes a different approach. We balance the need in our objective function for both few edges overall and low degrees for individual vertices by simultaneously limiting which vertices can buy edges and what edges they can buy.

[1] In [9], it was shown that this algorithm gives a 3-spanner for $\alpha = 1/3$, which already gives an $\tilde{O}(\|G\|_2^{1/3})$-approximation via Lemma 15. However, this is weaker than our final $\tilde{O}(\|G\|_2^{5/16})$ guarantee.

Algorithm 2 Edge/Vertex Sampling.

- For every vertex $v \in V$, and for every edge $e \in E$, independently sample uniformly random variables $z_v^- \in_R [0,1]$, $z_v^+ \in_R [0,1]$, and $z_e \in_R [0,1]$.
- For every edge $e = (u,v) \in E$, add e to the spanner if at least one of the following three conditions holds:
  1. $z_e \le x_e^{1/4}$ and $z_v^- \le x_e^{1/4}$.
  2. $z_u^- \le x_e^{1/4}$ and $z_v^+ \le x_e^{1/4}$.
  3. $z_u^+ \le x_e^{1/4}$ and $z_e \le x_e^{1/4}$.

▶ Remark 16. The algorithm is formulated for directed graphs. If the graph is undirected, run the algorithm on the directed graph where every original edge is considered with both possible orientations.

Both showing that these algorithms give a good approximation, and showing that together they give a valid 3-spanner, requires a technically involved argument. We separate these two arguments in the next two subsections.

In Section 3.2.2, we will show that every edge has a probability of $\Omega(1/\text{polylog}n)$ of being spanned by at least one of the two algorithms. Thus, the complete algorithm will be

- For some constant $c > 0$, run both Algorithm 1 and Algorithm 2 $O(\log^c n)$ times, and output the union of all the edges chosen by either algorithm over the various iterations.

Thus, for an approximation guarantee of $\tilde{O}(f)$, it suffices to show that the probability that either algorithm returns a subgraph with ℓ_2 norm greater than $\tilde{O}(\text{OPT} \cdot f)$ is at most $O(1/\log^c n)$ for some sufficiently large constant $c > 0$. This approximation guarantee (for $f = \|G\|_2^{5/16}$) is given in Section 3.2.1.

3.2.1 Approximation guarantee

As mentioned earlier, Lemma 15 implies that Algorithm 1 returns a subgraph with ℓ_2 norm at most $\text{OPT} \cdot \tilde{O}(\|G\|_2^{2/7})$ with probability at least $1 - 2^{-(1-o(1))\log^2 n}$. This is in fact better than our final approximation guarantee, so we will focus now on Algorithm 2.

We give the following upper bound on the ℓ_2 norm of the subgraph given by Algorithm 2.

▶ **Lemma 17.** *For any $b = b(n) > 1$, Algorithm 2 outputs a graph with ℓ_2 norm at most $\text{OPT} \cdot \tilde{O}(b^{1/2}\|G\|_2^{5/16})$ with probability at least $1 - \exp(-\Omega(\log^2 n)) - 1/b$.*

Proof. We will bound the contribution to the ℓ_2 norm of every kind of edge added by the algorithm. In particular, we define the three corresponding edge sets

$$E_1 = \left\{(u,v) \in E \mid z_{(u,v)} \le x_{(u,v)}^{1/4},\ z_v^- \le x_{(u,v)}^{1/4}\right\}$$
$$E_2 = \left\{(u,v) \in E \mid z_u^- \le x_{(u,v)}^{1/4},\ z_v^+ \le x_{(u,v)}^{1/4}\right\}$$
$$E_3 = \left\{(u,v) \in E \mid z_u^+ \le x_{(u,v)}^{1/4},\ z_{(u,v)} \le x_{(u,v)}^{1/4}\right\}$$

Now consider the various degrees defined by these edge sets:

$$d_1(u) = |\{v \in V \mid (u,v) \in E_1\}| \qquad d_2(v) = |\{u \in V \mid (u,v) \in E_1\}|$$
$$d_3(u) = |\{v \in V \mid (u,v) \in E_2\}| \qquad d_4(v) = |\{u \in V \mid (u,v) \in E_2\}|$$
$$d_5(u) = |\{v \in V \mid (u,v) \in E_3\}| \qquad d_6(v) = |\{u \in V \mid (u,v) \in E_3\}|$$

To bound the ℓ_2 norm of the subgraph returned by the algorithm, we bound each of $\sum_{u\in V}(d_i(u))^2$ seperately for each $i \in [6]$. However, we only analyze the contribution for $i = 1$ and $i = 3$. The analysis for $i = 6$ is identical to the analysis for $i = 1$, and the analysis for $i \in \{2, 4, 5\}$ is essentially identical to the analysis for $i = 3$.

Let us start by analyzing $\sum_{u\in V}(d_1(u))^2$. Note that for every $u \in V$, $d_1(u)$ is a sum of independent Bernoulli random variables with success probabilities

$$\text{Prob}[z_{(u,v)} \le x_{(u,v)}^{1/4}] \cdot \text{Prob}[z_v^- \le x_{(u,v)}^{1/4}] = x_{(u,v)}^{1/2}.$$

Thus, individual degrees behave exactly as in Edge-Round(1/2). Therefore, the proof of Lemma 15 (which did not use any property of the correlation between different degrees) shows that with high probability the total contribution to the ℓ_2 norm from these degrees is at most $\text{OPT} \cdot \tilde{O}(\|G\|_2^{1/4})$ (which is even smaller than our claim).

Now let us analyze the contribution from the d_3 degrees. First, for every vertex $u \in V$, let us define

$$\hat{\Gamma}(u) := \left\{ v \in V \mid (u, v) \in E,\ z_v^+ \le x_{(u,v)}^{1/4} \right\}.$$

Note, as before, that $|\hat{\Gamma}(u)|$ is a sum of independent Bernoulli random variables with probabilities $x_{(u,v)}^{1/4}$, and so with high probability, using Hölder's inequality, we have

$$|\hat{\Gamma}(u)| \le \log^2 n \cdot \sum_{v:(u,v)\in E} x_{u,v}^{1/4} \le \log^2 n \cdot d_G(u)^{3/4} d_{\text{LP}}(u)^{1/4}.$$

We will also need to bound (in expectation and with high probability) the following expression:

$$\begin{aligned}\mathbb{E}\left[\sum_{v\in\hat{\Gamma}(u)} x_{(u,v)}^{1/4}\right] &= \sum_{v:(u,v)\in E} x_{(u,v)}^{1/2} \\ &\le d_G(u)^{1/2} d_{\text{LP}}(u)^{1/2} \qquad \text{(by Cauchy-Schwarz).}\end{aligned}$$

This is not a sum of Bernoulli random variables, but it is a sum of independent random variables distributed in $[0, 1]$, where the expectation of the sum is at least 1, so we can use Theorem 13 and get that with probability at least $1 - 2^{-\log^2 n}$ we have

$$\sum_{v\in\hat{\Gamma}(u)} x_{(u,v)}^{1/4} \le \log^2 n \cdot d_G(u)^{1/2} d_{\text{LP}}(u)^{1/2}.$$

Suppose we have sampled the z^+ variables, so the sets $\hat{\Gamma}(u)$ are fixed and the two bounds on $\hat{\Gamma}(u)$ above hold for all $u \in V$. Note that this high probability event is completely independent of the z^- variables. Conditioned on this event, for every $u \in V$, $(d_3(u))^2$ is a random variable distributed in $[0, |\hat{\Gamma}(u)|^2]$ that depends only on z_u^-. The expected contribution from a single vertex $u \in V$ is

$$\begin{aligned}\mathbb{E}[(d_3(u))^2] &= \sum_{v_1,v_2\in\hat{\Gamma}(u)} \text{Prob}[z_u^- \le x_{(u,v_1)}^{1/4},\ z_u^- \le x_{(u,v_2)}^{1/4}] \\ &\le \sum_{v_1\in\hat{\Gamma}(u)} \sum_{v_2\in\hat{\Gamma}(u)} x_{(u,v_2)}^{1/4} \\ &= |\hat{\Gamma}(u)| \cdot \sum_{v\in\hat{\Gamma}(u)} x_{(u,v)}^{1/4} \\ &\le \log^4 n \cdot d_G(u)^{5/4} d_{\text{LP}}(u)^{3/4} \qquad \text{by our bounds on } \hat{\Gamma}(u)\end{aligned}$$

Thus, by Markov's inequality, with probability at least $1 - 1/b$ we have

$$\begin{aligned}
\sum_{u \in V} \mathbb{E}[d_3(u)^2] &\leq b \log^4 n \cdot \sum_{u \in V} d_G(u)^{5/4} d_{\mathrm{LP}}^{3/4} \\
&\leq b \log^4 n \|(d_G(u)^{5/4})_{u \in V}\|_{8/5} \|d_{\mathrm{LP}}(u)^{3/4}\|_{8/3} \qquad \text{by Hölder's inequality} \\
&= b \log^4 n \left(\sum_{u \in V} d_G(u)^2 \right)^{5/8} \left(\sum_{u \in V} d_{\mathrm{LP}}(u)^2 \right)^{3/8} \\
&= b \log^4 n \cdot \mathrm{LP}^{3/4} \|G\|_2^{5/4} \\
&\leq b \log^4 n \cdot \mathrm{OPT}^{3/4} \|G\|_2^{5/4} \\
&\leq b \ \mathrm{polylog}(n) \cdot \mathrm{OPT}^{3/4} \cdot \frac{\mathrm{OPT}^{5/4}}{\|G\|_2^{5/8}} \cdot \|G\|_2^{5/4} \qquad \text{by Theorem 2} \\
&= \tilde{O}(b \|G\|_2^{5/8}) \cdot \mathrm{OPT}^2
\end{aligned}$$

Thus the contribution of the d_3 degrees to the ℓ_2 norm is at most $\tilde{O}(\sqrt{b}\|G\|_2^{5/16}) \cdot \mathrm{OPT}$, as claimed. ◀

3.2.2 Correctness

We will use the following regularization lemma (simplified form of Lemma 2.6 in [9]):

▶ **Lemma 18.** *There exists a constant $C > 0$ such that for any vertices $u, v \in V$ and set P of paths from u to v of length at most 3 such that $\sum_{p \in P} y_p = 1$, there exists a subset $P' \subseteq P$ satisfying the following conditions.*

- *For some $1 \leq k \leq 3$, all paths in P' have length k.*
- *There exists some $y_0 > 0$ such that every path $p \in P'$ has weight $y_p \in [y_0, 2y_0]$. Furthermore, $1 \geq y_0|P'| \geq 1/(\log^C n)$.*
- *If $k = 3$, then all the paths in P' are tuples in $E_1 \times E_2 \times E_3$ for some pairwise disjoint collection of edge sets $E_1, E_2, E_3 \subset E$.*
- *If $k = 3$, then there exist positive integers d_L, d_R such that:*
  - *For every edge $e_1 \in E_1$, the number of paths in P' which include e_1 is in the range $[d_L, d_L \log^C n]$. Note that this gives $|E_1| \leq |P'|/d_L \leq 1/(d_L y_0)$.*
  - *Every edge $e_2 \in E_2$ participates in exactly one path in P'.*
  - *For every edge $e_3 \in E_3$, the number of paths in P' which include e_3 is in the range $[d_R, d_R \log^C n]$. Note that this gives $|E_3| \leq |P'|/d_R \leq 1/(d_R y_0)$.*

Note that given a solution to our LP relaxation, for every edge $(u, v) \in E$ there does exist such a set of paths P, and so there exists a set of paths P' of length k as in the lemma. If $k = 1$, this is just the edge (u, v) which then has LP value $y_{(u,v)} \geq 1/(\log^C n)$, and will be added by Algorithm 1 w.p. $\tilde{\Omega}(1)$. It is also easy to see that Algorithm 1 will span (u, v) if $k = 2$. Indeed, we have:

▶ **Lemma 19.** *Let P' be a set of paths of length k for an edge $(u, v) \in E$ as in Lemma 18. Then if $k = 2$, then w.p. $\tilde{\Omega}(1)$ at least one of these paths will be added by Algorithm 1.*

Proof. Since the paths in P' are of length 2, they are also edge disjoint. By the capacity constraints, every edge e in these paths must have LP value $x_e \geq y_0$. Thus, the probability that at least one path in P' will be added by Algorithm 1 is bounded by

$$\begin{aligned}
\text{Prob[Some path in } P' \text{ is added]} &= 1 - \prod_{p \in P'} \text{Prob[P' is not added]} \\
&= 1 - \prod_{p=(e_1,e_2)\in P'} (1 - \text{Prob}[e_1 \text{ is added}]\text{Prob}[e_2 \text{ is added}]) \\
&\geq 1 - \prod_{p \in P'} \left(1 - \left(y_0^{3/7}\right)^2\right) \\
&= 1 - \left(1 - y_0^{6/7}\right)^{|P'|} \\
&\geq 1 - \exp\left(-y_0^{6/7}|P'|\right) \\
&\geq 1 - \exp\left(1/\log^C n\right) = \tilde{\Omega}(1),
\end{aligned}$$

where the last inequality follows from the fact that $y_0^{6/7}|P'| \geq y_0|P'| \geq 1/\log^C n$. ◀

Thus it remains to deal with edges for which P' is a set of paths of length 3. We will show that every such edge is spanned with probability $\Omega(1)$ by either Algorithm 1 or Algorithm 2, depending on the parameters d_L, d_R from Lemma 18. In both cases, we show that the relevant algorithm adds a large number of paths from P' in expectation, and that outside some easy special cases, the number of paths added is (at least mildly) concentrated around the expectation by Chebyshev's inequality. The following lemma describes the correctness property of Algorithm 1.

▶ **Lemma 20.** *Let $(u, v) \in E$ be an edge and P' be a set of paths of length 3 as in Lemma 18 with corresponding parameters y_0, d_L, d_R. Then if $\max\{d_L, d_R\} \geq y_0^{-2/3}/\log^{C'} n$ for some constant $C' > 0$, Algorithm 1 will add at least one path in P' with probability $\tilde{\Omega}(1)$.*

Proof. First, consider the case where $d_L = \tilde{\Omega}(1/y_0)$. In this case, every edge $e_1 = (u, u') \in E_1$ that participates in any path in P' has LP value $x_{e_1} = \tilde{\Omega}(1)$. Thus, such an edge is added by Algorithm 1 w.p. $\tilde{\Omega}(1)$. Moreover, there are $\tilde{\Omega}(1/y_0)$ paths of length 2 from u' to v with LP value at least y_0 (suffixes of paths in P' starting with e_1), and by Lemma 19, w.p. $\tilde{\Omega}(1)$ Algorithm 1 will add at least one of these, and this event is independent of e_1 being added. Thus, in this case the lemma follows. The lemma follows similarly when $d_R = \tilde{\Omega}(1/y_0)$.

Assume therefore that for some arbitrarily large constant $C'' > 0$ we have

$$d_L, d_R \leq 1/(y_0 \log^{C''} n). \tag{4}$$

Now assume w.l.o.g. that $d_L > d_R$. Thus, by our assumption, we have

$$d_L \geq y_0^{-2/3}/\log^{C'} n. \tag{5}$$

Consider the case where $(d_L \leq) d_L d_R \leq y_0^{-2/3} \log^{C''} n$, which in particular implies $d_R \leq \log^{C''+C'} n$. In this case define a new set of paths $P'' \subseteq P'$ by taking for every edge $e_3 \in E_3$ a single path in P' containing e_3. Note that $|P''| \geq |P'|/(d_R \log^C n)$. Since $|E_1| \leq |P'|/d_L$ and every edge in E_1 participates in at most $d_L \log^C n$ paths in P', this implies that at least $|P'|/(2d_L d_R \log^{2C} n)$ edges in E_1 each participate in at least $d_L/(2d_R \log^C n)$ paths in $|P''|$. Let $E_1' \subseteq E_1$ be this set of edges. So we have

$$|E_1'| \geq \frac{|P'|}{2d_L d_R \log^{2C} n} \geq \frac{1}{2d_L d_R y_0 \log^{3C} n} \geq \frac{1}{2y_0^{1/3} \log^{3C+C''} n}.$$

Now, for every $e_1 \in E_1'$, denote by $P''(e_1)$ the set of paths in P'' containing e_1. By definition of E_1', for every such e_1 we have $|P''(e_1)| \geq d_L/(2d_R \log^C n)$. Note that by capacity constraints we have $x_{e_1} \geq d_L y_0 \geq y_0^{1/3}/\log^{C'} n$. Thus, as in the proof of Lemma 19, the probability that at least one path in $P''(e_1)$ is added is bounded by

$$\begin{aligned}
&\text{Prob[Some path in } P''(e_1) \text{ is added]} \\
&\quad = \text{Prob}[e_1 \text{ is added}] \cdot \left(1 - \prod_{p \in P''(e_1)} (1 - \text{Prob}[p \setminus e_1 \text{ is added}])\right) \\
&\quad \geq (d_L y_0)^{3/7} \left(1 - (1 - y_0^{6/7})^{|P''(e_1)|}\right) \\
&\quad \geq y_0^{1/7} \log^{-3C'/7} n \left(1 - (1 - y_0^{6/7})^{d_L/(2d_R \log^C n)}\right) \\
&\quad \geq y_0^{1/7} \log^{-3C'/7} n \left(1 - (1 - y_0^{6/7})^{y_0^{-2/3}/(2\log^{C+C''-C'} n)}\right) \\
&\quad \geq y_0^{1/7} \log^{-3C'/7} n \left(1 - \exp(-y_0^{4/21}/(2\log^{C+C''-C'} n))\right) \\
&\quad = \tilde{\Omega}(y_0^{1/3})
\end{aligned}$$

By definition of P'', the paths in $P''(e_1)$ are completely edge disjoint from the paths in $P''(e_1')$ for any $e_1, e_1' \in E_1'$. Thus, there is a probability of $\tilde{\Omega}(y_0^{1/3})$ that at least one path in $P''(e_1)$ is added, and these are independent events for the different edges $e_1 \in E_1'$. By our bound on $|E_1'|$, there are $\tilde{\Omega}(y_0^{-1/3})$ such edges, and so the probability that at least one of them will contribute a path in P'' to the spanner is at least $\tilde{\Omega}(1)$. This concludes our analysis of the case where $d_L d_R \leq y_0^{-2/3} \log^{C''} n$. From this point on, we assume that

$$d_L d_R \geq y_0^{-2/3} \log^{C''} n \tag{6}$$

Now define new LP values for $E_1 \cup E_2 \cup E_3$ as follows:

$$x'_e = \begin{cases} d_L y_0 & \text{if } e \in E_1 \\ y_0 & \text{if } e \in E_2 \\ d_R y_0 & \text{if } e \in E_3. \end{cases}$$

Then by capacity constraints, for every edge $e \in E_1 \cup E_2 \cup E_3$, we have $x_e \geq x'_e$. Consider an algorithm that adds edges $e \in E_1 \cup E_2 \cup E_3$ independently with probability $(x'_e)^{3/7}$ instead of $x_e^{3/7}$. Clearly, the probability that Algorithm 1 adds at least one path from P' is at least the probability that the algorithm with modified LP values does so. Let us now analyze the probability that the modified algorithm adds at least one path from P', and denote by Y the number of paths from P' added by the modified algorithm. We start by analyzing the expectation of Y. By definition of the modified algorithm, every path $p \in P'$ is added with probability

$$\text{Prob}[p \text{ is added}] = (d_L y_0)^{3/7} y_0^{3/7} (d_R y_0)^{3/7} = (d_L d_R)^{3/7} y_0^{9/7}.$$

Since $|P'| = y_0^{-1}/(\log^c n)$ for some $c \in [0, C]$, for an appropriate choice of C'', the expected number of paths added by the modified algorithm satisfies

$$\begin{aligned}
\mathbb{E}[Y] &= |P'|(d_L d_R)^{3/7} y_0^{9/7} \\
&= \log^{-c} n \cdot (d_L d_R)^{3/7} y_0^{2/7} && (7) \\
&\geq \log^{3C''/7 - c} n && \text{by (6)} \\
&\geq \log^{1+c/2+3C/2} n,
\end{aligned}$$

where the last inequality follows if we choose, say,

$$C'' = 7/3 + 7c/2 + 7C/2.$$

Thus, the expected number of paths in P' added is (relatively) large. However, since the paths in P' are not disjoint, this does not guarantee that at least one path will be added with probability $\tilde{\Omega}(1)$. To show this, we need to show concentration. As in [9], we use Chebyshev's inequality, which guarantees that $\text{Prob}[Y = 0] < \frac{1}{2}$ as long as

$$\text{Var}[Y] = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 < \frac{1}{8}(\mathbb{E}[Y])^2. \tag{8}$$

To show that (8) holds, consider the contribution of edge-disjoint versus non-edge-disjoint paths:

$$\begin{aligned}
\mathbb{E}[Y^2] &= \sum_{p_1,p_2 \in P'} \text{Prob}[p_1 \text{ and } p_2 \text{ are added}] \\
&= \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 \neq \emptyset}} \text{Prob}[p_1 \text{ and } p_2 \text{ are added}] + \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \emptyset}} \text{Prob}[p_1 \text{ is added}]\text{Prob}[p_2 \text{ is added}] \\
&\leq \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 \neq \emptyset}} \text{Prob}[p_1 \text{ and } p_2 \text{ are added}] + \sum_{p_1,p_2 \in P'} \text{Prob}[p_1 \text{ is added}]\text{Prob}[p_2 \text{ is added}] \\
&= \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 \neq \emptyset}} \text{Prob}[p_1 \text{ and } p_2 \text{ are added}] + \mathbb{E}[Y]^2
\end{aligned}$$

Thus, to show (8), it suffices to bound the contribution from pairs of non-edge-disjoint paths. These fall into three categories: pairs of identical paths, pairs sharing only the first edge (in E_1), and pairs sharing only the third edge (in E_3). The contribution from paths in the first category is at most

$$\sum_{p_1 = p_2 \in P'} \text{Prob}[p_1 \text{ and } p_2 \text{ are added}] = \mathbb{E}[Y] = o(\mathbb{E}[Y]^2). \qquad (\text{since } \mathbb{E}[Y] > \log n)$$

The analysis for the second and third categories is identical, so let us focus only on pairs of paths sharing only the first edge. These pairs contribute

$$\begin{aligned}
\sum_{e_1 \in E_1} \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \{e_1\}}} \text{Prob}[p_1 \text{ and } p_2 \text{ are added}] &= \sum_{e_1 \in E_1} \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \{e_1\}}} (d_L y_0)^{3/7} y_0^{6/7} (d_R y_0)^{6/7} \\
&= \sum_{e_1 \in E_1} \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \{e_1\}}} d_L^{3/7} d_R^{6/7} y_0^{15/7} \\
&\leq |E_1| (d_L \log^C n)^2 \cdot d_L^{3/7} d_R^{6/7} y_0^{15/7} \\
&\leq \frac{1}{d_L y_0} \cdot (\log^{2C} n) d_L^{17/7} d_R^{6/7} y_0^{15/7} \\
&= (\log^{2C} n) d_L^{10/7} d_R^{6/7} y_0^{8/7} \\
&\leq (\log^{2C - 4C''/7} n) d_L^{6/7} d_R^{6/7} y_0^{4/7} && \text{(by (4))} \\
&= \log^{2C + 2c - 4C''/7} n \cdot \mathbb{E}[Y]^2 && \text{(by (7))} \\
&= \log^{-4/3} n \cdot \mathbb{E}[Y]^2
\end{aligned}$$

Thus, all three categories contribute at most $o(\mathbb{E}[Y]^2)$, and this concludes the proof. ◀

Finally, we examine edges that will be spanned by Algorithm 2. Note that the trade-off between the two algorithms has nothing to do with the approximation guarantee. In fact, at the parameter threshold between the two algorithms (where $\max\{d_L, d_R\} \approx y_0^{-2/3}$), both algorithms either give or could be easily modified to give a better approximation than $\|G\|_2^{5/16}$. The reason for trading off the two algorithms is that beyond the threshold, Algorithm 2 will still give a large number of spanning paths for each edge in expectation, but in reality will only span such an edge with very low probability (in which event it will span it with many paths). The following lemma gives the parameter regime for edges spanned by Algorithm 2.

▶ **Lemma 21.** *Let $(u, v) \in E$ be an edge and P' be a set of paths of length 3 as in Lemma 18 with corresponding parameters y_0, d_L, d_R. Then there exists a constant $C' > 0$ such that if $d_L, d_R \leq y_0^{-2/3}/\log^{C'} n$, Algorithm 2 will add at least one path in P' with probability $\tilde{\Omega}(1)$.*

Proof. First consider the case where $d_L d_R \leq \log^{C''} n$ for some constant $C'' > 0$. This is an easy special case, but also the tight case of our entire analysis. In this case, define a new set of paths $P'' \subseteq P'$ by taking for every edge $e_3 \in E_3$ a single path in P' containing e_3, and then out of these, choosing for every edge $e_1 \in E_1$ used in these paths a single path containing e_1. Note that $|P''| \geq |P'|/\log^{2C+C''} n$, and all the paths in P'' are edge disjoint. For any path $p = (u, u', v', v) \in P''$, we can bound the probability that p is added by Algorithm 1 by

$$\begin{aligned}\text{Prob}[p \text{ is added}] &\geq \text{Prob}[z_{(u,u')} \leq x_{(u,u')}^{1/4}] \cdot \text{Prob}[z_{u'}^- \leq x_{(u,u')}^{1/4}, x_{(u',v')}^{1/4}] \\ &\quad \cdot \text{Prob}[z_{v'}^+ \leq x_{(u',v')}^{1/4}, x_{(v',v)}^{1/4}] \cdot \text{Prob}[z_{(v,v')} \leq x_{(v,v')}^{1/4}] \\ &\geq \text{Prob}[z_{(u,u')} \leq y_0^{1/4}] \cdot \text{Prob}[z_{u'}^- \leq y_0^{1/4}] \cdot \text{Prob}[z_{v'}^+ \leq y_0^{1/4}] \cdot \text{Prob}[z_{(v,v')} \leq y_0^{1/4}] \\ &= y_0.\end{aligned}$$

Since the paths in P'' are completely edge-disjoint, and share no interior vertices, the above events are independent for the various paths, and so the probability that at least one path in P'' is added is at least

$$\begin{aligned}1 - (1 - y_0)^{|P''|} \geq 1 - \exp(-y_0|P''|) &\geq 1 - \exp(-y_0|P'|/\log^{2C+C''} n) \\ &\geq 1 - \exp(-1/\log^{3C+C''} n) = \tilde{\Omega}(1).\end{aligned}$$

Now consider the remaining case. That is, assume from now on that

$$d_L d_R \geq \log^{C''} n. \tag{9}$$

As in the proof of Lemma 20, consider the result of running Algorithm 2 with LP values $\{x'_e\}$ as defined in that proof. As before, modifying the LP values in such a way can only decrease the probability that one of the paths in P' will be chosen. Let Y be the following set of paths, added by Algorithm 2 using the modified LP values:

$$\begin{aligned}Y = \Big\{(u, u', v', v) \in P' \mid &\left(z_{(u,u')} \leq (x'_{(u,u')})^{1/4}\right) \wedge \left(z_{u'}^- \leq (x'_{(u,u')})^{1/4}, (x'_{(u',v')})^{1/4}\right) \\ &\wedge \left(z_{v'}^+ \leq (x'_{(u',v')})^{1/4}, (x'_{(v',v)})^{1/4}\right) \wedge \left(z_{(v,v')} \leq (x'_{(v,v')})^{1/4}\right)\Big\}\end{aligned}$$

Since $d_L, d_R \geq 1$, the probability that a single path $p \in P'$ is added to Y is exactly

$$(d_L y_0)^{1/4} \cdot \min\{(d_L y_0)^{1/4}, y_0^{1/4}\} \cdot \min\{y_0^{1/4}, (d_R y_0)^{1/4}\} \cdot (d_R y_0)^{1/4} = (d_L d_R)^{1/4} y_0.$$

Since $|P'| = y_0^{-1}/(\log^c n)$ for some $c \in [0, C]$, for an appropriate choice of C'', the expected number of paths added by the modified algorithm satisfies

$$\begin{aligned}\mathbb{E}[Y] &= |P'|(d_L d_R)^{1/4} y_0 = \log^{-c} n \cdot (d_L d_R)^{1/4} && (10)\\ &\geq \log^{C''-c} n && \text{by (9)}\\ &\geq \log n,\end{aligned}$$

where the last inequality follows if we choose $C'' = 1 + c$.

As before, it is not enough to show that the expected number of paths is large, we also need to show concentration. As in the proof of Lemma 20, it suffices to show that

$$\sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 \neq \emptyset}} \text{Prob}[p_1, p_2 \subseteq Y] = o\left((\mathbb{E}[Y])^2\right).$$

These pairs of non-edge-disjoint paths fall into three categories: pairs of identical paths, pairs sharing only the first edge (in E_1), and pairs sharing only the third edge (in E_3). As before, the contribution from identical paths is $\mathbb{E}[Y] = o((\mathbb{E}[Y])^2)$ (where the final bound follows since $\mathbb{E}[Y] \geq \log n$). Since the analysis for the second and third categories is essentially the same, we focus on pairs of paths sharing only the first edge. These pairs contribute

$$\begin{aligned}\sum_{e_1 \in E_1} \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \{e_1\}}} \text{Prob}[p_1, p_2 \subseteq Y] &= \sum_{e_1 \in E_1} \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \{e_1\}}} (d_L y_0)^{1/4} y_0^{1/4} y_0^{1/2} (d_R y_0)^{1/2}\\ &= \sum_{e_1 \in E_1} \sum_{\substack{p_1,p_2 \in P' \\ p_1 \cap p_2 = \{e_1\}}} d_L^{1/4} d_R^{1/2} y_0^{3/2}\\ &\leq \frac{1}{d_L y_0} \cdot (d_L \log^C n)^2 \cdot d_L^{1/4} d_R^{1/2} y_0^{3/2}\\ &= (\log^{2C} n) d_L^{5/4} d_R^{1/2} y_0^{1/2}\\ &\leq (\log^{2C-3C'/4} n) d_L^{1/2} d_R^{1/2} \qquad (\text{since } d_L \leq y_0^{-2/3}/\log^{C'} n)\\ &= \log^{2C+2c-3C'/4} n \cdot \mathbb{E}[Y]^2 \qquad (\text{by (10)})\\ &\leq \log^{-3/4} n \cdot \mathbb{E}[Y]^2,\end{aligned}$$

where the final bound follows if we choose $C' \geq 1 + 8C/3 + 8c/3$. Thus, all three categories contribute at most $o(\mathbb{E}[Y]^2)$, and this concludes the proof. ◀

4 Generalizations and Open Questions

While we provided approximation and hardness bounds for Minimum ℓ_2-Norm 3-Spanner, the true approximability still remains open. Perhaps more interesting, though, is the question of the more general Minimum ℓ_p-Norm k-Spanner problem. Some of our techniques easily extend to this more general setting, but some do not. The linear-programming based framework we use to analyze the greedy algorithm should basically work for other values of p and k, but the details become more complicated.

Recall that our strongest approximation algorithm (from Section 3) is a careful tradeoff between greedy, independent edge sampling (**Edge-Round**), and a combined vertex and edge sampling (Algorithm 2). Independent edge sampling (**Edge-Round**) can also be analyzed for other values of p and k, where the right α to use depends on the value of k (indeed, this is the main technique used by [9] for $p = \infty$, and correctness for other values of k

follows directly from [9]). Our more tailored algorithm (Algorithm 2), which combines edge and vertex sampling, seems harder to generalize for larger values of k. Algorithm 2 is a generalization of the ideas used for $k = 3, 4$ in the ℓ_1 case (due to [12, 7, 13]), and it is a fascinating open question to extend these techniques to larger stretch values. For stretch $k = 3$ and other values of p, Algorithm 2 can be reanalyzed with appropriate parameters and seems to give nontrivial guarantees. In general, designing and analyzing approximation algorithms for other values of p and k remains an exciting challenge which may require new algorithmic ideas.

With respect to hardness, our results in Appendix A already include other values of p. For larger stretch values, the basic construction can be extended by including "outer paths" in the same way as has been done for many other spanner hardness results ([15, 9] in particular).

References

1 Noga Alon, Yossi Azar, Gerhard J. Woeginger, and Tal Yadid. Approximation Schemes for Scheduling. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '97, 1997.

2 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9(1):81–100, 1993. doi:10.1007/BF02189308.

3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.

4 Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *J. ACM*, 45(1):70–122, January 1998. doi:10.1145/273865.273901.

5 Yossi Azar, Leah Epstein, Yossi Richter, and Gerhard J. Woeginger. All-Norm Approximation Algorithms. In Martti Penttonen and Erik Meineche Schmidt, editors, *Algorithm Theory — SWAT 2002*, 2002.

6 Nikhil Bansal and Kirk Pruhs. Server Scheduling in the L_p Norm: A Rising Tide Lifts All Boat. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 242–250, 2003.

7 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and Directed Steiner Forest. *Inf. Comput.*, 222:93–107, 2013. doi:10.1016/j.ic.2012.10.007.

8 Barun Chandra, Gautam Das, Giri Narasimhan, and José Soares. New Sparseness Results on Graph Spanners. In *Proceedings of the Eighth Annual Symposium on Computational Geometry*, SCG '92, pages 192–201, New York, NY, USA, 1992. ACM. doi:10.1145/142675.142717.

9 Eden Chlamtáč and Michael Dinitz. Lowest-Degree k-Spanner: Approximation and Hardness. *Theory of Computing*, 12(15):1–29, 2016. doi:10.4086/toc.2016.v012a015.

10 Eden Chlamtáč, Michael Dinitz, and Thomas Robinson. The Norms of Graph Spanners. In *Proceedings of the 46th International Colloquium Conference on Automata, Languages, and Programming*, ICALP '19, 2019.

11 Michael Dinitz, Guy Kortsarz, and Ran Raz. Label Cover Instances with Large Girth and the Hardness of Approximating Basic k-Spanner. *ACM Trans. Algorithms*, 12(2):25:1–25:16, December 2015. doi:10.1145/2818375.

12 Michael Dinitz and Robert Krauthgamer. Directed Spanners via Flow-based Linear Programs. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 323–332, New York, NY, USA, 2011. ACM. doi:10.1145/1993636.1993680.

13 Michael Dinitz and Zeyu Zhang. Approximating Low-stretch Spanners. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 821–840, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2884435.2884494.

14 Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. URL: http://www.cambridge.org/gb/knowledge/isbn/item2327542/.

15 Michael Elkin and David Peleg. The Hardness of Approximating Spanner Problems. *Theor. Comp. Sys.*, 41(4):691–729, December 2007. doi:10.1007/s00224-006-1266-2.

16 Daniel Golovin, Anupam Gupta, Amit Kumar, and Kanat Tangwongsan. All-Norms and All-L_p-Norms Approximation Algorithms. In *FSTTCS*, volume 2 of *LIPIcs*, pages 199–210. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.

17 Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2):89–112, 2004. Special Issue on the 18th Annual Symposium on Computational Geometry - SoCG2002. doi:10.1016/j.comgeo.2004.03.003.

18 Guy Kortsarz. On the Hardness of Approximating Spanners. *Algorithmica*, 30(3):432–450, 2001. doi:10.1007/s00453-001-0021-y.

19 S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. doi:10.1109/TIT.1982.1056489.

20 Ran Raz. A Parallel Repetition Theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. doi:10.1137/S0097539795280895.

A Hardness Results

Since the analysis is simpler in the directed setting, we follow [9] and begin with it. We will also prove hardness for the more general ℓ_p-norm version, and hardness when $p = 2$ will follow as a corollary. First, though, we give some notation and background necessary for the reduction.

A.1 Background: Min-Rep and Spanner Hardness

Our hardness bounds rely on the Min-Rep problem. In Min-Rep we are given a bipartite graph $G = (A, B, E)$ where A is partitioned into groups $A_1, A_2, \ldots, A_r$ and B is partitioned into groups $B_1, B_2, \ldots, B_r$, with the additional property that every set A_i and every set B_j has the same size (which we will call $|\Sigma|$ due to its connection to the alphabet of a 1-round 2-prover proof system). This graph and partition induces a new bipartite graph G' called the *supergraph* in which there is a vertex a_i for each group A_i and similarly a vertex b_j for each group B_j. There is an edge between a_i and b_j in G' if there is an edge in G between some node in A_i and some node in B_j. A node in G' is called a supernode, and similarly an edge in G' is called a superedge.[2]

A REP-cover is a set $C \subseteq A \cup B$ with the property that for all superedges $\{a_i, b_j\}$ there are nodes $a \in A_i \cap C$ and $b \in B_j \cap C$ where $\{a, b\} \in E$. We say that $\{a, b\}$ *covers* the superedge $\{a_i, b_j\}$. The goal is to construct a REP-cover of minimum size.

For any fixed constant $\epsilon > 0$, we say that an instance of Min-Rep is a YES instance if $OPT = 2r$ (i.e. a single node is chosen from each group) and is a NO instance if $OPT \geq 2^{\log^{1-\epsilon} n} r$. We will sometimes refer to the hardness gap (in this case $2^{\log^{1-\epsilon} n}$) as the *soundness* s, due to the connection between Min-Rep and proof systems. The following theorem is due to Kortsarz [18] (the polynomial relations between the parameters are implicit rather than explicit in his proof, but are straightforward to verify since the instances used in [18] are obtained by parallel repetition [20] applied to instances of 3SAT-5 which have a constant gap [4, 3]).

[2] Rather than G being the graph and G' being the supergraph, sometimes G' is referred to as the graph and G is called the *label-extended graph*.

▶ **Theorem 22** ([18]). *Unless* NP $\subseteq$ DTIME($2^{\text{polylog}(n)}$)*, for any constant* $\epsilon > 0$ *there is no polynomial-time algorithm that can distinguish between YES and NO instances of Min-Rep. This is true even when the graph and the supergraph are regular, and both the supergraph degree and* $|\Sigma|$ *are polynomial in the soundness.*

A.2 Directed Hardness

A.2.1 Reduction

We first consider the directed setting (note that here the "degree" in the degree vector is the sum of the in-degree and the out-degree). Suppose we are given a Min-Rep instance $\widetilde{G} = (A, B, \widetilde{E})$ with associated supergraph $G' = (U, V, E')$ from Theorem 22. For any vertex $w \in U \cup V$ we let $\Gamma(w)$ denote its group. So $\Gamma(u) \subseteq A$ for $u \in U$, and $\Gamma(v) \subseteq B$ for $v \in V$. Similarly, for $a \in A \cup B$, we let $\Gamma^{-1}(a)$ be the unique $w \in U \cup V$ such that $a \in \Gamma(w)$. We will assume without loss of generality that G' is regular with degree $d_{G'}$ and $\widetilde{G}$ is regular with degree $d_{\widetilde{G}}$. Our reduction will also use a special bipartite regular graph $H = (X, Y, E_H)$, which will simply be the directed complete bipartite graph with $|X| = |Y|$. Let d_H denote the degree of a node in H, so $d_H = |X| = |Y|$ (later when we move to the undirected setting H will not just be the complete bipartite graph). We will set all of these values to $(d_{G'} + |\Sigma| + 1)^{p/(p-1)}$ (for the undirected setting we will set d_H to this value, but $|X| = |Y|$ will be larger).

Our instance $G = (V_G, E_G)$ of Minimum ℓ_p-Norm 3-Spanner will be a combination of these three graphs. The four sets of vertices are

$$V^L_{out} = U \times X \qquad V^R_{out} = V \times Y$$
$$V^L_{in} = A \times E_H \qquad V^R_{in} = B \times E_H.$$

The actual vertex set V_G of our instance G will be $V^L_{out} \cup V^R_{out} \cup V^L_{in} \cup V^R_{in}$. Defining the edge set is a little more complex, as there are a few different types of edges. We first create *outer edges*, which are incident on outer nodes:

$$E_{out} = \{((u, x), (v, y)) : u \in U \wedge v \in V \wedge x \in X \wedge y \in Y \wedge \{u, v\} \in E' \wedge (x, y) \in E_H\}.$$

Note that if we fix x and y the corresponding outer edges form a copy of the supergraph G'. Thus these edges essentially form $|E_H|$ copies of the supergraph.

We also have *inner edges*, which correspond to $|E_H|$ copies of the Min-Rep instance (note that unlike the supergraph copies, these copies are vertex disjoint):

$$E_{in} = \{((a, e), (b, e)) : a \in A \wedge b \in B \wedge e \in E_H \wedge \{a, b\} \in \widetilde{E}\}.$$

We will now add *connection edges*, i.e., edges that connect some of the outer nodes to some of the inner nodes. Let

$$E^L_{con} = \{((u, x), (a, (x, y))) : u \in U \wedge a \in \Gamma(u) \wedge x \in X \wedge (x, y) \in E_H\}, \text{ and}$$
$$E^R_{con} = \{((b, (x, y)), (v, y)) : v \in V \wedge b \in \Gamma(v) \wedge y \in Y \wedge (x, y) \in E_H\}.$$

In other words, the outer node (u, x) (resp. (v, y)) is connected to the inner nodes in its group in each copy of $\widetilde{G}$ that corresponds to an E_H edge that involves x (resp. y).

Finally, for technical reasons we need to add *group edges* internally in each group in each copy of $\tilde{G}$: let $E^L_{group} = \{((a, e), (a', e)) : e \in E_H \wedge a, a' \in A \wedge \Gamma^{-1}(a) = \Gamma^{-1}(a')\}$, and let $E^R_{group} = \{((b, e), (b', e)) : e \in E_H \wedge b, b' \in B \wedge \Gamma^{-1}(b) = \Gamma^{-1}(b')\}$.

Our final edge set is the union of all of these, namely $E_G = E_{out} \cup E_{in} \cup E^L_{con} \cup E^R_{con} \cup E^L_{group} \cup E^R_{group}$.

A.2.2 Analysis

We first consider the YES case. We can use almost the same spanner as was used to prove the equivalent lemma in [9] (Lemma 3.3). Unfortunately, since in [9] only the maximum degree mattered, they did not need to optimize the degrees of non-extremal vertices, while we do. So we actually use a slightly sparser spanner construction.

▶ **Lemma 23.** *If $\widetilde{G}$ is a YES instance of Min-Rep, then there is a 3-spanner S of G with $\|S\|_p \leq 3d_H(|U||X| + |V||Y|)^{1/p}$*

Proof. Since $\widetilde{G}$ is a YES instance, for each $u \in U$ there is some $f(u) \in \Gamma(u)$ and for each $v \in V$ there is some $f(v) \in \Gamma(v)$ so that $\{f(u), f(v)\} \in \tilde{E}$ for all $\{u, v\} \in E'$. Our spanner S the connection edges suggested by the REP-cover: for every $u \in U$ and $x \in X$ and $(x, y) \in E_H$, it contains the connection edge $((u, x), (f(u), (x, y)))$. Similarly, for every $v \in V$ and $y \in Y$ and $(x, y) \in E_H$, it contains the connection edge $((f(v), (x, y)), (v, y))$. It also contains a star of group edges centered at the chosen node in every group: for every $u \in U$ and $e \in E_H$ and $a \in \Gamma(u)$ it includes the group edges $((f(u), e), (a, e))$ and $((a, e), (f(u), e))$, and for every $v \in V$ and $e \in E_H$ and $b \in \Gamma(v)$ it includes the group edges $((f(v), e), (b, e))$ and $((b, e), (f(v), e))$. Finally, it contains the appropriate inner edges: for every $\{u, v\} \in E'$ with $u \in U$ and $v \in V$ and every $e \in E_H$, we add the inner edge $((f(u), e), (f(v), e))$.

This is precisely the spanner from [9, Lemma 3.3] but with fewer group edges (we include stars in each group, while [9, Lemma 3.3] included all group edges). It is easy to verify that this change does not affect the correctness of the spanner: all edges in G not in S are still spanned. So we rely on [9, Lemma 3.3] for correctness.

So we just need to analyze $\|S\|_p$. To do this, we can just count the degrees in S of each type of nodes. There are $|U||X| + |V||Y|$ outer nodes, each of which has degree at most d_H in S. For the inner nodes, we divide into those that are chosen (those that are $(f(u), e)$ or $(f(v), e)$ for some u or v in $U \cup V$) and those that are not. There are at most $|E_H|(|A| + |B|)$ inner nodes which are not chosen, and in S they all have degree 2 (an incoming and outgoing group edge from the node in the same group that is chosen). There are at most $|E_H|(|U| + |V|)$ inner nodes which are chosen, each of which has degree in S of at most $|\Sigma| + d_{G'} + 1$ (the group edges, inner edges, and connection edges that it is incident with respectively). Putting all this together, we get that

$$\begin{aligned}
\|S\|_p &\leq ((|U||X| + |V||Y|) \cdot d_H^p \\
&\quad + |E_H|(|A| + |B|) \cdot 2^p + |E_H|(|U| + |V|)(|\Sigma| + d_{G'} + 1)^p)^{1/p} \\
&\leq ((|U||X| + |V||Y|) \cdot d_H^p + 2|E_H|(|U| + |V|)(|\Sigma| + d_{G'} + 1)^p)^{1/p} \\
&\leq (3(|U||X| + |V||Y|) \cdot d_H^p)^{1/p} \\
&\leq 3d_H(|U||X| + |V||Y|)^{1/p},
\end{aligned}$$

where we have used our setting of d_H and the fact that $|A| + |B| = (|U| + |V|)|\Sigma|$. ◀

Now we analyze the NO setting.

▶ **Lemma 24.** *If $\widetilde{G}$ is a NO instance of Min-Rep, then every 3-spanner S of G has $\|S\|_p \geq sd_H(|U||X| + |V||Y|)^{1/p}$.*

Proof. Suppose for the sake of contradiction that this is false. Let S be a 3-spanner of G with $\|S\|_p < sd_H(|U||X| + |V||Y|)^{1/p}$. For every outer node (u, x) in V_{out}^L and edge $(x, y) \in E_H$, let $d_{out}^{x,y}(u, x)$ be the number of outer edges in S that are incident on (u, x) and have the other

endpoint of the form (v, y) for some $v \in V$. Similarly, for every outer node (v, y) in V^R_{out} and edge $(x, y) \in E_H$, let $d^{x,y}_{out}(v, y)$ be the number of outer edges in S that are incident with (v, y) and have the other endpoint of the form (u, y). For every outer node (u, x) in V^L_{out} and edge $(x, y) \in E_H$, let $d^{x,y}_{con}(u, x)$ be the number of connection edges in S that are incident with (u, x) and have the other endpoint of the form $(a, (x, y))$ for some $a \in \Gamma(u)$. Similarly, for every outer node (v, y) in V^R_{out} and edge $(x, y) \in E_H$, let $d^{x,y}_{con}(v, y)$ be the number of connection edges in S that are incident with (v, y) and have the other endpoint of the form $(b, (x, y))$ for some $b \in \Gamma(v)$.

Now with this notation in hand, the fact that $\|S\|_p \leq sd_H(|U||X|+|V||Y|)^{1/p}$ implies that

$$\sum_{(x,y)\in E_H} \left(\sum_{u\in U} ((d^{x,y}_{out}(u,x))^p + (d^{x,y}_{con}(u,x))^p) + \sum_{v\in V} ((d^{x,y}_{out}(v,y))^p + (d^{x,y}_{con}(v,y))^p) \right)$$
$$\leq (sd_H)^p(|U||X| + |V||Y|)$$

Now a simple application of Hölder's inequality gives us the following.

$$\sum_{(x,y)\in E_H} \left(\sum_{u\in U} (d^{x,y}_{out}(u,x) + d^{x,y}_{con}(u,x)) + \sum_{v\in V} (d^{x,y}_{out}(v,y) + d^{x,y}_{con}(v,y)) \right)$$
$$\leq (sd_H)(|U||X| + |V||Y|) \qquad \text{(Hölder's inequality)}$$
$$\leq 2sd_H(|U||X|) \qquad (H \text{ and } G' \text{ are both balanced})$$
$$\leq |E_H|2s|U| \qquad (H \text{ is regular with degree } d_H)$$
$$\leq |E_H|s(|U| + |V|) \qquad (G' \text{ is balanced})$$

Thus averaging now implies that there is some $(x, y) \in E_H$ such that

$$\sum_{u\in U} (d^{x,y}_{out}(u,x) + d^{x,y}_{con}(u,x)) + \sum_{v\in V} (d^{x,y}_{out}(v,y) + d^{x,y}_{con}(v,y)) \leq s(|U| + |V|) \tag{11}$$

Fix this (x, y). We create a set $C_1(u) \subseteq \Gamma(u)$ for each $u \in U$ by adding all $a \in \Gamma(u)$ such that there is a connection edge $((u, x), (a, (x, y)))$ which contributes to $d^{x,y}_{con}(u, x)$. Similarly, we create a set $C_1(v) \subseteq \Gamma(v)$ for each $v \in V$ by adding all $b \in \Gamma(v)$ such that there is a connection edge $((b, (x, y)), (v, y))$ which contributes to $d^{x,y}_{con}(v, y)$.

Now we create similar sets for the outer edges. For each $u \in U$ we create a set $C_2(u) \subseteq \Gamma(u)$ and for each $v \in V$ we create a set $C_2(v)$ as follows. For every outer edge $((u, x), (v, y))$ in S (i.e., every outer edge which contributes to $d^{x,y}_{out}(u) + d^{x,y}_{out}(v)$), we pick an arbitrary $a \in \Gamma(u)$ and $b \in \Gamma(v)$ such that $\{a, b\} \in \widetilde{E}$ and add a to $C_2(u)$ and b to $C_2(v)$.

Let $C(u) = C_1(u) \cup C_2(u)$ for all $u \in U$, and let $C(v) = C_1(v) \cup C_2(v)$ for all $v \in V$. Let $C = (\cup_{u\in U} C(u))) \cup (\cup_{v\in V} C(v))$. Clearly by construction we know that

$$|C| \leq \sum_{u\in U} ((d^{x,y}_{out}(u,x)) + (d^{x,y}_{con}(u,x))) + \sum_{v\in V} ((d^{x,y}_{out}(v,y)) + (d^{x,y}_{con}(v,y)))$$
$$\leq s(|U| + |V|). \qquad \text{(by (11))}$$

Now we claim that C is a valid REP-cover. This will prove the lemma, since it will imply that S is not a NO instance, giving a contradiction and thus implying that no such S can exist. To see that C is a REP-cover, consider an arbitrary superedge $\{u, v\} \in E'$. It is not hard to see (and was proved in [9]) that the only way that S can span the outer edge $((u, x), (v, y))$ is to either include that edge in S or include a *canonical path* between the endpoints: a path which uses a connection edge to get to some $(a, (x, y))$, then an inner

edge to get to some $(b,(x,y))$, then a connection edge to get to (v,y). If the outer edge is included in S, then when we constructed $C_2(u)$ and $C_2(v)$ we explicitly added some $a \in \Gamma(u)$ and $b \in \Gamma(v)$ that cover $\{u,v\}$. Otherwise S spans the outer edge using a canonical path, which from the construction of $C_1(u)$ and $C_1(v)$ means that there is some $a,b \in C$ which covers $\{u,v\}$. Thus C is a REP-cover, which proves the lemma. ◀

Now we can prove our hardness bound using these lemmas.

▶ **Theorem 25.** *Unless* NP $\subseteq$ DTIME$(2^{\text{polylog}(n)})$*, for any constant* $\epsilon > 0$ *and* $p \geq 1$ *there is no polynomial-time algorithm that can approximate* Minimum ℓ_p-Norm 3-Spanner *in directed graphs better than* $2^{\left(\frac{p-1}{3p-1}\right)^{1-\epsilon}\log^{1-\epsilon}n}$.

Proof. Lemmas 23 and 24, together with Theorem 22, imply that under the complexity assumption, there is no polynomial-time algorithm with approximation ratio better than

$$\frac{sd_H(|U||X|+|V||Y|)^{1/p}}{3d_H(|U||X|+|V||Y|)^{1/p}} = \frac{s}{3}.$$

The only thing that remains is to argue about the increase in the size: the n in the value of s is really $|A|+|B|$, while our graph G is larger. But it is not too much larger: the number of vertices in G is $|V_G| = |U||X|+|V||Y|+|A||E_H|+|B||E_H| = O(n(|\Sigma|+d_{G'})^{2p/(p-1)}) \leq O(n^{1+\frac{2p}{p-1}}) = O(n^{(3p-1)/(p-1)})$. Thus the overall hardness that we obtain is

$$\frac{s}{3} = \frac{1}{3}2^{\log^{1-\epsilon}n} = \frac{1}{3}2^{\log^{1-\epsilon}(N^{(p-1)/(3p-1)})} = \frac{1}{3}2^{\left(\frac{p-1}{3p-1}\right)^{1-\epsilon}\log^{1-\epsilon}N}.$$

The extra $1/3$ factor can be absorbed by using a smaller ϵ. ◀

Our claimed hardness theorem for $p=2$, the directed version of Theorem 8, is a corollary of this theorem for $p=2$.

A.3 Undirected Hardness

We extend the directed hardness to the undirected setting in the same way that it was extended for LDkS in [9]. First, we start with a slightly different Min-Rep instance with some useful extra properties (from [11] instead of from [18], and with some extra analysis from [9]). Then we combine it with a graph H which is the finite projective plane of degree $d_H = (d_{G'}+|\Sigma|+1)^{p/(p-1)}$, which is a graph of girth 6 with $|X| = |Y| = d_H^2$. Then we further subsample G to ensure that there are no cycles of length less than 5 consisting of outer edges (some were introduced via the way we combined $\widetilde{G}$ with H). All of this is necessary in order to ensure that in any 3-spanner of G, the only ways of spanning an outer edge are through the outer edge itself or through a canonical path (and in particular, there is no way to span it using just other outer edges).

We give a sketch of the analysis and proof here, since it is simply re-analyzing the construction of [9] using the ideas from the previous section. It is straightforward to prove the analog of Lemma 23, since we use the same spanner suggested by the existence of a good REP-cover and analyze all degrees in the same way. This implies that in a YES instance, there will be a k-spanner S with $\|S\|_p \leq O((|U||X|+|V||Y|)^{1/p} \cdot d_H)$.

The NO setting is more difficult to analyze, since it requires arguing directly about the subsampling process. But if we follow the analysis in [9] but with the notation from the previous section, we get that in a NO instance,

$$\sum_{u \in U}(d^{x,y}_{out}(u,x) + d^{x,y}_{con}(u,x)) + \sum_{v \in V}(d^{x,y}_{out}(v,y) + d^{x,y}_{con}(v,y)) \geq s^{1/8}(|U|+|V|)$$

for every $\{x, y\} \in E_H$. This is the equivalent of Equation (11) but as a direct proof rather than by contradiction. Then as in the directed case, we can combine these to get the following theorem (the dependence on p is slightly worse since the graph that we build is larger due to using the finite projective plane rather than the complete bipartite graph).

▶ **Theorem 26.** *Unless* NP $\subseteq$ BPTIME($2^{\text{polylog}(n)}$), *for any constant* $\epsilon > 0$ *and* $p \geq 1$ *there is no polynomial-time algorithm that can approximate* Minimum ℓ_p-Norm 3-Spanner *better than* $2^{\left(\frac{p-1}{4p-1}\right)^{1-\epsilon} \log^{1-\epsilon} n}$.

Theorem 8 is now a corollary of this theorem when $p = 2$.