Lowest Degree k-Spanner: Approximation and Hardness

Eden Chlamtáč[1] and Michael Dinitz[2]

1 **Ben Gurion University, IL**
`chlamtac@cs.bgu.ac.il`

2 **Johns Hopkins University, U.S.**
`mdinitz@cs.jhu.edu`

Abstract

A k-spanner is a subgraph in which distances are approximately preserved, up to some given stretch factor k. We focus on the following problem: Given a graph and a value k, can we find a k-spanner that minimizes the maximum degree? While reasonably strong bounds are known for some spanner problems, they almost all involve minimizing the total number of edges. Switching the objective to the degree introduces significant new challenges, and currently the only known approximation bound is an $\tilde{O}(\Delta^{3-2\sqrt{2}})$-approximation for the special case when $k = 2$ [Chlamtáč, Dinitz, Krauthgamer FOCS 2012] (where Δ is the maximum degree in the input graph). In this paper we give the first non-trivial algorithm and polynomial-factor hardness of approximation for the case of general k. Specifically, we give an LP-based $\tilde{O}(\Delta^{(1-1/k)^2})$-approximation and prove that it is hard to approximate the optimum to within $\Delta^{\Omega(1/k)}$ when the graph is undirected, and to within $\Delta^{\Omega(1)}$ when it is directed.

1998 ACM Subject Classification G.2.2 Graph Theory: Graph algorithms

Keywords and phrases Graph spanners, approximation algorithms, hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2014.80

1 Introduction

A spanner of a graph is a sparse subgraph that approximately preserves distances. Formally, a k-spanner of a graph $G = (V, E)$ is a subgraph H of G in which $d_H(u, v) \leq k \cdot d_G(u, v)$ for all $u, v \in V$, where d_H and d_G denote shortest path distances in H and G, respectively[1]. Graph spanners were originally introduced in the context of distributed computing [22, 23], and since then have been extensively studied from both a distributed and a centralized perspective. Much of this work has focused on the fundamental tradeoffs between stretch, size, and total weight, such as the seminal result of Althöfer et al. that every graph admits a $(2k-1)$-spanner with at most $n^{1+1/k}$ edges [1] and its many extensions (e.g. to dealing with total weight [7]). Spanners have also appeared as fundamental building blocks in a wide range of applications, from routing in computer networks [25] to property testing of functions [4].

In parallel with this work on the fundamental tradeoffs there has been a line of work on approximating spanners. In this setting we are usually given an input graph G and a stretch value k, and our goal is to construct the best possible k-spanner. If "best" is measured in terms of the total number of edges, then clearly the construction of [1] gives

[1] Equivalently, a subgraph H is a k-spanner if $d_H(u, v) \leq k$ for every edge (u, v) in G.

17th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'14) / 18th Int'l Workshop on Randomization and Computation (RANDOM'14).
Editors: Klaus Jansen, José Rolim, Nikhil Devanur, and Cristopher Moore; pp. 80–95
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

an $O(n^{2/(k+1)})$-approximation (for odd k), simply because $\Omega(n)$ is a trivial lower bound on the size of any spanner of a connected graph. However, when the objective function is to minimize the maximum degree, there are no non-trivial fundamental bounds like there are for the number of edges, so it is natural to consider the optimization problem. Moreover, degree objectives are notoriously difficult (consider degree-bounded minimum spanning trees [24] as opposed to general minimum spanning trees), and so almost all work on approximation algorithms for spanners has focused on minimizing the number of edges, as opposed to maximum degree.

We call the problem of minimizing the degree of a k-spanner the Lowest Degree k-Spanner problem (which we will abbreviate LDkS). For directed graphs, the degree is the sum of the in- and out-degrees.[2] Kortsarz and Peleg initiated the study of the maximum degree of a spanner, giving an $O(\Delta^{1/4})$-approximation for LD2S [21] (where Δ is the maximum degree of the input graph). This was only recently improved to $\tilde{O}(\Delta^{3-2\sqrt{2}+\epsilon})$ for arbitrarily small $\epsilon > 0$ by Chlamtáč, Dinitz, and Krauthgamer [10]. The only known hardness for LD2S was $\Omega(\log n)$ [21]. Despite the length of time since minimizing the degree was first considered (over 15 years) and the significant amount of work on other spanner problems, no nontrivial upper or lower bounds were known previous to this work for LDkS when $k \geq 3$.

1.1 Our results and techniques

We give the first nontrivial upper and lower bounds for the approximability of Lowest Degree k-Spanner for $k \geq 3$. We assume throughout that all edges have length 1; while much previous work has dealt with spanners with arbitrary edge lengths, our results (and all previous results on optimizing the degree) are specific to uniform edge lengths. Handling general edge lengths is an intriguing open problem.

As we also note later, it is easy to see that any k-spanner must have maximum degree at least $\Delta^{1/k}$ (simply to span the edges incident to the node of maximum degree). Thus, simply outputting the original graph is a $\Delta^{1-1/k}$ approximation. We beat the trivial algorithm, and give the following algorithmic result[3]:

▶ **Theorem 1.** *For any integer $k \geq 1$, there is an $\tilde{O}(\Delta^{(1-\frac{1}{k})^2})$-approximation for* Lowest Degree *k-*Spanner.

While this may seem like a rather small improvement over the trivial $\Delta^{1-1/k}$-approximation, it still requires significant technical work (possibly explaining why no nontrivial bounds were known previously). Note that in the special case of $k = 2$ our bound recovers the bound of [21], although not the improved one of [10]. This is not a coincidence: our algorithm is a modification of [21], albeit with a very different and significantly more involved analysis. We use a natural flow-based linear program in which the decision variable for each edge is interpreted as a capacity, while the spanning requirement is interpreted as requiring that for every original edge $\{u, v\}$ there is enough capacity to send 1 unit of flow along paths of length at most k (this is essentially the same LP used for directed spanners by [12, 4] but with a degree objective, and reduces to the LP used by [21] when $k = 2$).

[2] With appropriate changes to the LP, our algorithm also works for the variant in which we measure the out-degree.

[3] Our algorithm and analysis work for both the undirected and directed case with no change. The parameter k is taken to be a constant, and the $\tilde{O}$ notation hides polylogarithmic factors of the form $O(\log n(\log \Delta)^c)$ for some $c = c(k)$.

The LP rounding in [21] was a simple independent randomized rounding which ensured that every path of length 2 is contained in the spanner with probability that is at least the LP flow along that path. Since paths of length 2 with common endpoints are naturally edge-disjoint, these events are independent (for a fixed edge (u, v)), and a simple calculation shows that at least one $u - v$ path survives the rounding with probability at least $1 - 1/e$.

When $k \geq 3$ the structure of these paths becomes significantly more complicated. While we still guarantee that each flow path will be contained in the spanner with probability proportional to the amount of flow in the path, we can no longer guarantee independence, as the flow paths are not disjoint, and may intersect and overlap in highly non-trivial ways. Our main technical contribution (in the upper bound) shows that the rounding exhibits a certain dichotomy: either we can carefully prune the paths (while retaining $1/\text{polylog}(\Delta)$ flow) until they are disjoint, or the number of flow-paths that survive the rounding is concentrated around an expectation which is $\omega(1)$. This ensures that (after boosting by repeating the rounding a polylogarithmic number of rounds), every edge is spanned with high probability.

On the lower bound side, our main result is the following:

▶ **Theorem 2.** *For any integer $k \geq 3$, there is no polynomial time algorithm that can approximate* Lowest Degree k-Spanner *better than $\Delta^{\Omega(1/k)}$ unless* $\text{NP} \subseteq \text{BPTIME}(2^{\text{polylog}(n)})$.

We can actually get a stronger hardness result if we assume that the input graph is directed:

▶ **Theorem 3.** *There is some constant $\gamma > 0$ such that for any integer $k \geq 3$ there is no polynomial time algorithm that can approximate* Lowest Degree k-Spanner *on directed graphs better than Δ^{γ}, unless* $\text{NP} \subseteq \text{BPTIME}(2^{\text{polylog}(n)})$.

It is important to note that these hardness results do not hold if we replace Δ by n, as the algorithmic results do. The instances generated by the hardness reduction have a maximum degree that is subpolynomial in n, so the best hardness that we would be able to prove (in terms of n) would be subpolynomial (although still superpolylogarithmic). On the other hand, by phrasing the hardness in terms of Δ we not only allow direct comparisons to the upper bounds, but also allow us to use techniques (namely reductions from Label Cover and Min-Rep) that typically give only subpolynomial hardness results. Our hardness results require a mix of previous techniques and ideas, but with some interesting twists.

There is a well-developed framework (mostly put forward by Kortsarz [19] and Elkin and Peleg [15]) for proving hardness for spanner problems by reducing from Min-Rep, a minimization problem related to Label Cover that has proven useful for proving hardness (see Section 3 for the formal definition). Our reductions have two key modifications. First, we boost the degree by including many copies of both the starting Min-Rep instance and the added gadget nodes. This was unnecessary for previous spanner problems because boosting the degree was not necessary – it was sufficient to boost the number of edges by including many copies of just the gadget nodes.

The second modification is particular to the undirected case. Undirected spanner problems are difficult to prove hard because if we try to simply apply the generic framework for reducing from Min-Rep, there can be extra “fake" paths that allow the spanner to bypass the Min-Rep instance altogether. Elkin and Peleg [16] showed that for basic (min-cardinality rather than min-degree) undirected k-spanner it was sufficient to use Min-Rep instances with large girth: applying the framework to those instances would yield hardness for basic k-spanner. But they left open the problem of actually proving that Min-Rep with large girth was hard. This was proved recently [11] by subsampling the Min-Rep instance to get rid of short cycles while still preserving hardness., finally proving hardness for basic k-spanner.

We might hope LDkS is similar enough to basic k-spanner that we could just apply the generic reduction to Min-Rep with large girth. Unfortunately this does not work, since the steps we take to boost the degree end up introducing short cycles even if the starting Min-Rep instance has large girth (unlike the reduction used for basic k-spanner [16]). So we might instead hope that we could simply use the *ideas* of [11], and subsample after doing the reduction rather than before. Unfortunately this does not work either. Instead, we must do both: apply the normal reduction to the special (already subsampled) Min-Rep instances from [11], and then do an extra, separate round of subsampling on the reduction. In other words, we must sample both the Min-Rep instance itself *and* the graph obtained by applying the generic reduction to these already sampled instances.

1.2 Related Work

There has been a huge amount of work on graph spanners, from their original introduction in the late 80's [22, 23] to today. The best bounds on the tradeoff between stretch and space were reached by Althöfer et al. [1].

Most of the work since then has been on extending these tradeoffs (e.g. including additive stretch [2, 9], fault-tolerance [8, 13], or average stretch [6]) or considering algorithmic aspects such as allowing fast distance queries [26] or extremely fast constructions [18].

In parallel with this, there has been a line of work on approximating graph spanners. This was initiated by Kortsarz and Peleg, who gave an $O(\log(|E|/|V|))$-approximation for the sparsest 2-spanner problem [20] and then an $O(\Delta^{1/4})$-approximation for Lowest Degree 2-Spanner [21]. This was followed by upper bounds by Elkin and Peleg [17] for a variety of related spanner problems including LD2S (although not LDkS).

With the exception of [21], one feature that the approximation algorithms for spanners have shared with the global bounds on spanners has been the use of purely combinatorial techniques. Kortsarz and Peleg introduced the use of linear programming for spanners [21], but this was a somewhat isolated example. More recently, linear programming relaxations have become a dominant technique, and have been used for transitive closure spanners [5], directed spanners [12, 4], fault-tolerant spanners [12, 13], and LD2S [10]. In this paper we use a rounding scheme similar to [21] (with a much more complicated analysis) and an LP that is a degree-based variant of the flow-based LP introduced by [12] (an earlier use of flow-based LPs for approximating spanners is [14]).

On the hardness side, the first results were due to Kortsarz [19] who proved $\Omega(\log n)$-hardness for the basic k-spanner problem (for constant k) and $2^{\log^{1-\epsilon} n}$-hardness for a weighted version. These results were pushed further by Elkin and Peleg [15], who proved the same $2^{\log^{1-\epsilon} n}$-hardness for a collection of spanner problems including directed k-spanner. Separately, Kortsarz and Peleg proved logarithmic hardness for LD2S [21]. Proving strong hardness for basic k-spanner remained open until recently, when Dinitz, Kortsarz, and Raz proved it by showing that Min-Rep is hard even when the instances have large girth [11]. They accomplished this through careful subsampling, which we push further by subsampling both before and after the reduction.

1.3 Preliminaries

We now give some basic formal definitions which will be useful throughout this paper. Given an unweighted graph $G = (V, E)$, we let $d_G(u, v)$ denote the shortest-path distance from u to v in G, i.e. the minimum number of edges in any path from u to v (note that if G is directed this may be asymmetric). The *girth* of a graph is the minimum number of edges in any cycle in the graph. We use the notation $e \sim v$ to indicate that e is incident on v, and the notation $p : u \rightsquigarrow v$ to indicate that p is a path from u to v. We think of paths as tuples of edges, and denote by $(p)_i$ the ith edge in a path p. For integer k, we will use $[k]$ to denote the set $\{1, 2, \ldots, k\}$.

A *k-spanner* of G is a subgraph H of G in which $d_H(u, v) \leq k \cdot d_G(u, v)$ for all $u, v \in V$. The value k is referred to as the *stretch* of the spanner. The fundamental problem that we are concerned with is the following:

▶ **Definition 4.** Suppose we are given an unweighted graph G and a stretch parameter k. The problem of computing the k-spanner that minimizes the maximum degree is Lowest Degree k-Spanner.

2 The algorithm

We now present our approximation algorithm for LDkS, proving Theorem 1. It is not hard to see that a subgraph with maximum degree D can only be a k-spanner if the original graph has degree at most $\sum_{i=1}^{k} D^i = (1 + o(1))D^k$ (the maximum number of possible paths of length $\leq k$ starting from a given node in the spanner). Therefore, we have

▶ Observation 5. In a graph with maximum degree Δ, any k-spanner must have maximum degree at least $\Omega(\Delta^{1/k})$.

2.1 LP relaxation, rounding, and approximation guarantee

Our algorithm uses the following natural LP relaxation:

$$\begin{aligned}
\min \quad & d \\
\text{s.t.} \quad & \textstyle\sum_{e \sim v} x_e \leq d && \forall v \in V && (1) \\
& \textstyle\sum_{p: u \rightsquigarrow v, |p| \leq k} y_p = 1 && \forall (u,v) \in E && (2) \\
& x_e \geq \sum_{\substack{p: u \rightsquigarrow v, |p| \leq k \\ p \ni e}} y_p && \forall (u,v), e \in E && (3) \\
& x_e, y_p \geq 0 && \forall e, p && (4)
\end{aligned}$$

Note that this LP has polynomial size when k is constant, and can even be solved in polynomial time when k is superconstant [12].

Recall that in a k-spanner, it is sufficient to span every edge by a path of length k. Note that for any edge $(u, v) \in E$ there may be multiple paths in the spanner spanning this edge. However, we can always pick one such path per edge. In the intended (integral) solution to the above formulation, x_e is an indicator for whether e appears in the spanner, and y_p is an indicator for the unique spanner path we assign to (u, v) (it could even be just the edge itself, if $p = (u, v)$). Thus, combined with the above observation, we have

▶ Observation 6. In a graph with maximum degree Δ, in which the optimal solution to the above LP is d_{LP}, any k-spanner (including the optimum spanner) must have maximum degree at least $\Omega(\max\{d_{\text{LP}}, \Delta^{1/k}\})$.

We apply a rather naïve rounding algorithm to the LP solution, which can be thought of as a natural extension of the rounding in [21] for LD2S:

- Independently add each edge $e \in E$ to the spanner with probability $x_e^{1/k}$.

The heart of our analysis is showing that in the subgraph this produces, every original edge is spanned with probability at least $\tilde{\Omega}(1)$. It is then only a matter of repeating the above algorithm a polylogarithmic number of times to ensure that every edge is spanned w.h.p. This will only incur a polylogarithmic factor in the degree guarantee. The following lemma gives an easy bound on the expected degree of any vertex in the above rounding:

▶ **Lemma 7.** *Let H be the subgraph obtained from the above rounding, and let d_{OPT} be the smallest possible degree of a k-spanner of $G=(V,E)$. Then every vertex in H has expected degree at most $O(d_{\mathrm{OPT}}\Delta^{(1-1/k)^2})$.*

Proof. By linearity of expectation, the expected degree of any $v \in V$ is

$$\begin{aligned}\textstyle\sum_{e\sim v} x_e^{1/k} &\le (\deg_G(v))^{1-1/k}\left(\textstyle\sum_{e\sim v} x_e\right)^{1/k} && \text{by Jensen's inequality}^4\\ &\le \Delta^{1-1/k} d_{\mathrm{LP}}^{1/k}\\ &= O\left(d_{\mathrm{OPT}}\Delta^{1-1/k}\cdot\frac{d_{\mathrm{LP}}^{1/k}}{\max\{d_{\mathrm{LP}},\Delta^{1/k}\}}\right) && \text{by Observation 6}\end{aligned}$$

Noting that the last expression is maximized when $d_{\mathrm{LP}}=\Delta^{1/k}$, we get

$$\textstyle\sum_{e\sim v} x_e^{1/k} = O(d_{\mathrm{OPT}}\Delta^{1-1/k}\cdot(\Delta^{1/k})^{1/k-1}) = O(d_{\mathrm{OPT}}\Delta^{(1-1/k)^2}).$$

◀

▶ Remark. Note that a simple Chernoff bound says that all degrees will be concentrated around their respective expectations, as long as the expectations are sufficiently large (say $\ge 3\ln n$). Since we repeat the basic algorithm at least $3\ln n$ times, the concentration argument can be applied to the total number of incident edges added, with multiplicities.

Thus, the crux of the analysis is to show that, indeed, every edge will be spanned with some reasonable probability.

2.2 Sketch of proof of correctness

Suppose, for simplicity, that for an edge $(u,v)\in E$, all the contribution in (2) (the spanning constraint) comes from paths of length exactly k. First, suppose all the paths with non-zero weight y_p in (2) are disjoint. For every edge e in such a path p we have, from (3), that $x_e \ge y_p$. Therefore, the probability that such a path survives (i.e. all the edges in it are retained in the rounding) is $\prod_{e\in p} x_e^{1/k} \ge y_p$. Denoting by $P(=P(u,v))$ the set of such paths, by disjointness these events are independent, and therefore we have

$$\text{Prob}[(u,v)\text{ is spanned}] \ge 1-\prod_{p\in P}(1-y_p) \ge 1-\prod_{p\in P} e^{-y_p} = 1-e^{\sum_{p\in P} y_p} = 1-1/e.$$

Thus, repeating this process $O(\log n)$ times, all such edges will be spanned w.h.p.

However, $u \rightsquigarrow v$ paths of length ≥ 3 need not be disjoint in general. We may assume that all paths $p\in P$ have some fixed length $k' \in [k]$ and are tuples of the form $(e_i)_{i\in I}\in \prod_{i\in I} E_i$ for some disjoint edge sets $E_1,\dots,E_k \subset E$ (see Lemma 8). Consider the extreme example where $k'=k$ and the flow is distributed evenly over all possible paths of the form $u - v_1 - v_2 - \ldots - v_{k-1} - v$ for $v_i \in V_i$, where $\{V_i \mid i\in[k]\}$ is an equipartition of $V\setminus\{u,v\}$.

[4] or Hölder's inequality

Here, the amount of flow through each edge in the first and last layers is roughly $(k-1)/n$, and the amount of flow through any edge in the other layers is roughly $((k-1)/n)^2$. Thus, in the worst case, edges in the first and last layers will have values $x_e = (k-1)/n$ and in the other layers $x_e = ((k-1)/n)^2$. It is easy to see that the number of edges from u to V_1 that are still present (after the rounding) is concentrated around $(n/(k-1))^{1-1/k}$ (since each outgoing edge from u is retained independently with probability $((k-1)/n)^{1/k}$). Similarly, every vertex in layers $i = 2, \ldots, k-2$ will retain $\sim (n/(k-1))^{1-2/k}$ edges to the next layer, creating a total of $(n/(k-1))^{1-1/k+(1-2/k)(k-2)}$ paths from u to V_{k-1}, an $(n/(k-1))^{-1/k}$ fraction of which will continue to v. Thus, not only is (u,v) spanned after the rounding, it is spanned by $\sim (n/(k-1))^{(k^2-3k+2)/k}$ different paths (unlike the disjoint case, where only a constant number of paths survive).

Thus intuitively we have two scenarios: either the paths are disjoint, or they overlap, and a large number of them survive (both in expectation and w.h.p. due to concentration). However, this is not easy to formalize (moreover, we note that on an edge-by-edge basis, gradually merging two paths does not monotonically increase the probability that at least one path survives). To greatly simplify the formalization of this dichotomy, we prune the paths to achieve near-regularity in the LP values and combinatorial structure of the flow. To describe the outcome of the pruning, we need to introduce one more notation: Given a set of paths P' and (small) set of edges S, we denote by $m_{P'}(S)$ the number of paths $p \in P'$ such that p contains S. For example, $m'_P(\emptyset) = |P'|$ and for any path $p \in P'$ (considering p as a set of edges), $m_{P'}(p) = 1$.

The pruning procedure, which is only needed for the analysis, is an extension of standard pruning techniques (e.g. pruning to make a bipartite graph nearly regular), and is summarized in the following lemma, whose proof will appear in the full version of this paper.

▶ **Lemma 8.** *There exists a function f such that for any vertices $u, v \in V$ and set P of paths from u to v of length at most k such that $\sum_{p\in P} y_p \geq 1/\text{polylog}(\Delta)$, there exists a subset of paths $P' \subseteq P$ satisfying:*

- *For some $k' \in [k]$, all paths in P' have length k'.*
- *All the paths in P' are tuples in $\prod_{i=1}^{k'} E_i$ for some pairwise disjoint collection of sets $E_1, \ldots, E_{k'} \subset E$.*
- *There exists some $y_0 > 0$ such that every path has weight $y_p \in [y_0, 2y_0]$. Furthermore, $y_0|P'| \geq 1/(\log \Delta)^{f(k)}$.*
- *There exists a positive integer vector $(m_I)_{I\subseteq k'}$ such that $m_{P'}((e_i)_{i\in I}) \in [m_I, m_I(\log \Delta)^{f(k)}]$ for every index set $\emptyset \neq I \subseteq [k']$ and every I-tuple $(e_i)_{i\in I} \in \prod_{i\in I} E_i$ which is contained in some path in P'. (Note that if $e \in (e_i)$ then $m((e_i)) \leq m(e)$ and therefore $m_I \leq m_i(\log \Delta)^{f(k)}$ for $i \in I$).*

We note that if $\prod_{i=1}^{k'} m_{\{i\}} \leq \text{polylog}(\Delta)$, this is quite close to the disjoint paths case (where $m_{\{i\}} = 1$), and can be analyzed accordingly. The following Lemma gives the relevant result for this case.

▶ **Lemma 9.** *Let P' be the set of paths given by Lemma 8, and suppose $\prod_{i=1}^{k'} m_{\{i\}} < (\log \Delta)^{g(k)}$. Then with probability at least $1/(\log \Delta)^{h(k)}$ (for some function h), at least one path in P' survives the rounding.*

Proof. For the sake of the analysis, let us prune the paths even further. Go through every level E_i for $i = 1, \ldots, k'$ sequentially, and for every $e \in E_i$, choose exactly one (undeleted) path that contains e and delete all other paths containing e. Since for all $e \in E_i$

we have $m_{P'}(e) \leq m_{\{i\}}(\log \Delta)^{f(k)}$, in each level we retain at least a $1/(m_{\{i\}}(\log \Delta)^{f(k)})$-fraction of paths. Therefore, we end up with a new collection of paths $P^* \subseteq P'$ such that $|P^*| \geq |P'|/(\log \Delta)^{g(k)+k'f(k)}$, and the paths in P^* are edge-disjoint.

The analysis is now straightforward. Every path $p \in P^*$ is retained with probability

$$\prod_{e\in p} x_e^{1/k} \geq \prod_{e \in p} y_p^{1/k} \geq y_0^{k'/k} \geq (|P'|(\log \Delta)^{f(k)})^{-k'/k}$$

There are $|P^*|$ such paths, and each survives independently of the rest, therefore, at least one path in P^* survives with probability

$$\begin{aligned}1 - (1 - \prod_{i=1}^{k'} x_i^{1/k})^{|P^*|} &\geq 1 - \exp(-|P^*| \prod_{i=1}^{k'} x_i^{1/k}) \\ &\geq 1 - \exp(-|P'|^{(k-k')/k}(\log \Delta)^{-(f(k)k'/k+g(k)+k'f(k))}) \\ &\geq 1 - \exp(-(\log \Delta)^{-(1+1/k)(f(k)+g(k))}) \\ &= (1 - o(1))(\log \Delta)^{-(1+1/k)(f(k)+g(k))}.\end{aligned}$$

◀

We can also easily deal with the case $m_{\{i\}} \geq |P'|/\text{polylog}(\Delta)$, which indicates that in some layer i, the paths are concentrated in a small number of edges, by choosing just one edge $e \in E_i$, contracting this edge, and deleting all paths that do not use e (see the proof of Theorem 11). Thus, the main case we have to deal with is the intermediate case, where there is non-negligible overlap ($\prod m_{\{i\}}$ is not too small), but also no edges have too large a load (no $m_{\{i\}}$ is too close to $|P'|$). It is not hard to show that in this case the expected number of paths will be large, but showing concentration is more challenging. This constitutes the bulk of the technical analysis.

To briefly describe this part of the analysis, consider a single edge $e \in E_i$. We know this edge is contained in $m(\{e\})$ paths in P', and each of these paths has weight $y_p \in [y_0, 2y_0]$. Therefore, by constraint (3) and Lemma 8, we have

$$x_e \geq m(\{e\})y_0 \geq m_{\{i\}}y_0 \geq m_{\{i\}}/(|P'|(\log \Delta)^{f(k)}). \tag{5}$$

Suppose instead of sampling each edge independently with probability $x_e^{1/k}$, we retained any edge $e \in E_i$ with probability $x_i^{1/k}$ for

$$x_i := m_{\{i\}}/(|P'|(\log \Delta)^{f(k)}),$$

and let Y be the number of paths in P' that survive this rounding. This is clearly a lower bound for the number of paths retained in our original rounding algorithm (we can think of the modified rounding as first applying the original rounding, and then subsampling the edges even further). Note that $\mathbb{E}[Y] = |P'|^{1-k'/k}(\prod_i m_{\{i\}})^{1/k'}/(\log \Delta)^{f(k)k'/k}$, so, as we've mentioned, if $\prod_i m_{\{i\}}$ is large, then $\mathbb{E}[Y]$ will also be large. By Chebyshev's inequality, we can bound the probability that $Y = 0$ by

$$\text{Prob}[Y = 0] \leq \text{Prob}[Y < \mathbb{E}[Y]/2] \leq \text{Prob}\left[(Y - \mathbb{E}[Y])^2 > \tfrac{1}{4}(\mathbb{E}[Y])^2\right] < \frac{4\text{Var}[Y]}{(\mathbb{E}[Y])^2}$$

Thus, to prove, say, that $\text{Prob}[Y = 0] < \frac{1}{2}$, it suffices to show that

$$(\text{Var}[Y] =)\ \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 < \frac{1}{8}(\mathbb{E}[Y])^2. \tag{6}$$

While the proof of this bound is somewhat technical, it is greatly simplified by the pruning phase, which allows us to bound the variance directly as a function of the m_I values without having to analyze the combinatorial structure of the flow. The result for the main case is given by the following lemma, whose proof is deferred to the full version due to space constraints:

▶ **Lemma 10.** *Let P' be the set of paths given by Lemma 8. Then if $\prod_{i=1}^{k'} m_{\{i\}} \geq (\log\Delta)^{g(k)}$, and for every $i \in [k']$ we have $m_{\{i\}} \leq |P'|(\log\Delta)^{-g(k)}$, where $g(k) \geq (4k+2)f(k)$ then (6) holds.*

Finally, we combine these three components to give our correctness guarantee:

▶ **Theorem 11.** *Let P' be a collection of $u \rightsquigarrow v$ paths as in Lemma 8, then with probability at least $1/(\log\Delta)^{l(k)}$ (for some function l), at least one path in P' survives the rounding.*

Proof. First, consider the case where $m_{\{i\}} \leq |P'|(\log\Delta)^{-(4k+2)f(k)}$ for every $i \in [k']$. In this case, if $\prod_{i=1}^{k'} m_{\{i\}} \geq (\log\Delta)^{(4k+2)f(k)}$, then the theorem follows directly from our second moment argument and Lemma 10. If $\prod_{i=1}^{k'} m_{\{i\}} < (\log\Delta)^{(4k+2)f(k)}$, on the other hand, then the theorem follows from Lemma 9.

On the other hand, if there does exist some $i \in [k']$ such that $m_{\{i\}} \geq |P'|(\log\Delta)^{-(4k+2)f(k)}$, then the above analysis breaks down. In this case, choose any edge $e \in E_i$, and note that (by (5))

$$x_e \geq m_{\{i\}}/(|P'|(\log\Delta)^{f(k)}) \geq (\log\Delta)^{-(4k+3)f(k)}.$$

Suppose $e = (s,t)$. In the undirected case, we have a minor technical detail: we choose the direction (s,t) or (t,s) which contains at least $x'_e/2$ flow in P, say (s,t). Let P'_e be the set of paths in P' that use (s,t) (in this direction) (in the directed case, P'_e is just the set of paths in P' that use e). Then every path in P'_e consists of three parts: a $u \rightsquigarrow s$ prefix of length $i-1$, the edge (s,t), and a $t \rightsquigarrow v$ suffix of length $k'-i$. Let P''_e be the set of contracted paths $\{p/\{e\} \mid p \in P'_e\}$ in the contracted graph $G/\{e\}$. The paths in P''_e are clearly in a one-to-one correspondence with the paths in P'_e. Note that the paths P''_e satisfy all the properties given by Lemma 8 with $(4k+4)f(k)$ in place of $f(k)$ (where we define $m_{P''_e}(S) := m_{P'}(S \cup \{e\})$).

The original rounding will retain edge e with probability at least $(\log\Delta)^{-(4+3/k)f(k)}$. However, by induction on k, there is also a $(\log\Delta)^{-l(k-1)}$ probability that some (contracted) $u \rightsquigarrow v$ path in P''_e will survive. Since this event is independent of the event where e is retained, we have that at least one path in P'_e will survive with probability at least $(\log\Delta)^{-(4+3/k)f(k)-l(k-1)}$. ◀

Applying the above theorem to the set P' of paths given by Lemma 8 applied to the set P of all $u \rightsquigarrow v$ paths of length at most k, it follows that (one iteration of) our rounding algorithm spans every edge with reasonably large probability:

▶ **Corollary 12.** *Given a solution to the LP relaxation, our rounding algorithm spans every edge (by a path of length at most k) with probability at least $1/\text{polylog}(\Delta)$.*

3 Hardness of Approximation

Our reductions are based on the framework developed by [19, 15]. 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.

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.

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.

▶ **Theorem 13** ([19]). *Unless* $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{\mathrm{polylog}(n)})$*, 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.*

In the basic reduction framework we start with a Min-Rep instance, and then for every group we add a vertex (corresponding to the supernode) which is connected to vertices in the group using paths of length approximately $k/2$. We then add an edge between any two supernodes that have a superedge in the supergraph. So there is an "outer" graph corresponding to the supergraph, as well as an "inner" graph which is just the Min-Rep graph itself. The basic idea is that the only way to span a superedge is to use a path of length k that goes through the Min-Rep instance, in which case the Min-Rep edge that is in this path corresponds to nodes in a valid REP-cover. So if we are in a YES instance there is a small REP-cover and thus a small spanner, while if we are in a NO instance every REP-cover is large and thus the spanner must have many edges in order to span the superedges.

In [15] and [19] this framework is used to prove hardness of approximation when the objective is the number of edges by creating many copies of the outside nodes (i.e. the supergraph), all of which are connected to the same inner nodes (Min-Rep graph). This forces the number of edges used in the spanner to essentially equal the size of a valid REP-cover, as all other edges used by the spanner become lower order terms. We reverse this, by creating many copies of the inner Min-Rep graph. If we simply connect a single copy of the outer graph we run into a problem, though: each superedge can be spanned by paths through *any* of the copies. There is nothing that forces it to be spanned through *all* of them, and thus nothing that forces degrees to be large. We show how to get around this by creating many copies of both the inner and the outer graph, but using asymptotically more copies of the inner graph than the outer.

3.1 Directed LDkS

We now consider the directed setting, but due to space constraints only give an outline.

Suppose we are given a bipartite Min-Rep instance $\widetilde{G} = (A, B, \widetilde{E})$ with associated supergraph $G' = (U, V, E')$. 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 $v \in V$. 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 $d_H = |X| = |Y|$. We will set all of these values to $d_{G'} + 2|\Sigma| + 1$.

Our LDkS instance $G = (V_G, E_G)$ will be a combination of these three graphs. Let $k_L = \lfloor \frac{k-1}{2} \rfloor$, and let $k_R = \lceil \frac{k-1}{2} \rceil$. The four sets of vertices are

$$V_{out}^L = U \times X \times [k_L] \qquad V_{out}^R = V \times Y \times [k_R]$$
$$V_{in}^L = A \times E_H \qquad V_{in}^R = B \times E_H.$$

The actual vertex set V_G of our LDkS instance G will be $V_{out}^L \cup V_{out}^R \cup V_{in}^L \cup V_{in}^R$. We say that an outer node is *maximal* if its final coordinate is maximal (k_L for nodes in V_{out}^L or k_R for nodes in V_{out}^R), and we say that an outer node is *minimal* if its final coordinate is 1.

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 maximal outer nodes:

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

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 edges that connect some of the outer nodes to some of the inner nodes: let

$$E_{con}^L = \{((u,x,1),(a,(x,y))) : u \in U \wedge a \in \Gamma(u) \wedge x \in X \wedge (x,y) \in E_H\}, \text{ and}$$
$$E_{con}^R = \{((b,(x,y)),(v,y,1)) : v \in V \wedge b \in \Gamma(v) \wedge y \in Y \wedge (x,y) \in E_H\}.$$

In other words, the minimal outer node for each (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).

We now need to connect the minimal outer nodes and the maximal outer nodes. We do this by creating paths: let

$$E_{path}^L = \{((u,x,i),(u,x,i-1)) : u \in U, x \in X, i \in \{2,\dots k_L\}\}, \text{ and}$$
$$E_{path}^R = \{((v,y,i),(v,y,i+1)) : v \in V, y \in Y, i \in [k_R - 1]\}.$$

Finally, for technical reasons we need to add edges internally in each group in each copy of $\tilde{G}$: let $E_{group}^L = \{((a,e),(a',e)) : e \in E_H \wedge a, a' \in \Gamma(u) \text{ for some } u \in U\}$, and let $E_{group}^R = \{((b,e),(b',e)) : e \in E_H \wedge b, b' \in \Gamma(v) \text{ for some } v \in V\}$.

Our final edge set is the union of all of these, namely $E_{out} \cup E_{in} \cup E_{con}^L \cup E_{con}^R \cup E_{path}^L \cup E_{path}^R \cup E_{group}^L \cup E_{group}^R$.

3.1.1 Analysis

The first step is to show that if there is a small REP-cover for the original Min-Rep instance, then there is a k-spanner with low maximum degree. To do this we will use the notion of a *canonical path* for an outer edge. Consider an outer edge $((u,x,k_L),(v,y,k_R))$. A path from (u,x,k_L) to (v,y,k_R) is *canonical* it if includes $k_L - 1$ path edges, followed by a connection edge, an inner edge, another connection edge, and then $k_R - 1$ path edges. Note that any such path has length $k_L + k_R + 1 = k$, so can be used to span the outer edge. Furthermore, note that any such path corresponds to selecting two nodes (the inner nodes hit by the path) that cover the $\{u,v\}$ superedge in the original Min-Rep instance.

It is not hard to see that the *only* way to span an outer edge is either through a canonical path (which corresponds to a way of covering the associated superedge in the Min-Rep instance) or including the edge itself. This means that we can span all outer edges by using canonical paths corresponding to a REP-cover, and that this is the only way spanning outer edges. Since in a YES instance there is a REP-cover in which only a single node is selected per group, we can use those canonical paths to construct a k-spanner with maximum degree at most d_H.

▶ **Lemma 14.** *If we start with a YES instance of Min-Rep, then there is a k-spanner of G which has maximum degree at most $d_H + 1$.*

Proof. Since we are in 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 contains all edges in E^L_{group} and E^R_{group} as well as all edges in E^L_{path} and E^R_{path}. It also contains 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 $((u, x, 1), (f(u), (x, y)))$. Similarly, for every $v \in V$ and $y \in Y$ and $(x, y) \in E_H$, it contains the connection edge $((f(v), (x, y)), (v, y, 1))$. 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 $((f(u), e), (f(v), e))$.

In this spanner, the degree of outer nodes which are not minimal is at most 2 (the 2 incident path edges), and the degree of inner nodes is at most $d_{G'} + 2|\Sigma| + 1$ (since they are incident on one connection edge, $2|\Sigma|$ group edges, and $d_{G'}$ inner edges). The degree of a minimal outer node is at most $d_H + 1$, since it is incident on 1 path edge and for each edge incident on the second coordinate in E_H it is incident to a single inner node. Thus the maximum degree of the spanner is at most $\max\{d_{G'} + 2|\Sigma| + 1, d_H + 1\} = d_H + 1$ as claimed.

It remains to show that this is indeed a valid spanner. The only edges not included are the outer edges and some of the connection edges and inner edges, so we simply need to prove that they are spanned by paths of length at most k. For connection edges this is trivial. Consider some edge $((u, x, 1), (a, (x, y))) \in E^L_{con}$. Clearly there is a path of length two that spans it: an included connection edge from $(u, x, 1)$ to $(f(u), (x, y))$, followed by a group edge from $(f(u), (x, y))$ to $(a, (x, y))$. A similar path exists (in the opposite direction) for connection edges in E^R_{con}.

Similarly, consider an inner edge $((a, e), (b, e))$ which is not in the spanner. Let $u \in U$ and $v \in V$ so that $a \in \Gamma(u)$ and $b \in \Gamma(v)$. Then $\{u, v\} \in E'$, so our spanner contains an inner edge $((f(u), e), (f(v), e))$. So there is a path of length three in our spanner from (a, e) to (b, e), namely $(a, e) \to (f(u), e) \to (f(v), e) \to (b, e)$, where the first and last edges are group edges and the middle edge is an inner edge.

Now consider an outer edge $((u, x, k_L), (v, y, k_R))$. We can span it by using a canonical path, where the first connection edge will be from $(u, x, 1)$ to $(f(u), (x, y))$, the inner edge will be from $(f(u), (x, y))$ to $(f(v), (x, y))$, and the second connection edge will be from $(f(v), (x, y))$ to $(v, y, 1)$ (this fixes the path edges used as well). Note that all of these edges exist in the spanner, since the connection edges are included by construction and the inner edge must exist because this is a YES instance, i.e. because $\{f(u), f(v)\} \in \tilde{E}$ for all $\{u, v\} \in E'$. Thus this is indeed a path in the spanner, and it clearly has length k. ◀

On the other hand, since in a NO-instance there are no small REP-covers, any spanner must include either many canonical paths or many outer edges. This lets us prove that in this case every k-spanner has some node with large degree.

▶ **Lemma 15.** *If we start with a NO instance on Min-Rep, then every k-spanner of G has maximum degree at least $(s/3)d_H$*

Proof. We will prove the contrapositive, that if there is a k-spanner of G with maximum degree less than $(s/3)d_H$ then there is a REP-cover of size less than $s(|U| + |V|)$ (and thus we did not start with a NO instance). Let $\hat{G}$ be such a spanner. We create a bucket $B_{(x,y)}$ for each edge $(x, y) \in E_H$, which will contain a collection of outer edges and connection edges that are in $\hat{G}$. For each outer edge $((u, x, k_L), (v, y, k_R))$ that is in $E(\hat{G})$, we add it to the bucket $B_{(x,y)}$. Similarly, for each connection edge $((u, x, 1), (a, (x, y)))$ that is in $E^L_{con} \cap E(\hat{G})$ we add it to $B_{(x,y)}$, as well as each connection edge $((b, (x, y)), (v, y, 1)) \in E^R_{con} \cap E(\hat{G})$. Since $\hat{G}$ has maximum degree less than $(s/3)d_H$, the total number of edges in buckets (i.e. the total number of outer and connection edges in $\hat{G}$) is less than $|U||X|(s/3)d_H$ (the number of outer edges) plus $|U||X|(s/3)d_H + |V||Y|(s/3)d_H$ (the number of connection edges), for a total of $|U||X|sd_H$ edges (since both G' and H are balanced and regular).

Since H is regular we know that $|X|d_H = |E_H|$. Thus there must exist some bucket with less than $s|U| = s|V|$ edges. Let $B_{(x,y)}$ be this bucket. We will create a REP-cover as follows. For each edge $((u, x, 1), (a, (x, y))) \in E^L_{con} \cap B_{(x,y)}$ we will include a and for each edge $((b, (x, y)), (v, y, 1)) \in E^R_{con} \cap B_{(x,y)}$ we will include b. For each outer edge $((u, x, k_L), (v, y, k_R))$ we will include an arbitrary vertex in $\Gamma(u)$ and an arbitrary vertex in $\Gamma(v)$ that are adjacent in $\widetilde{G}$ (such vertices must exist in order for the Min-Rep instance to be satisfiable at all). Clearly this cover has size less than $2|B_{(x,y)}| \leq 2s|U| = s(|U| + |V|)$.

It only remains to show that this is a valid cover. To see this, consider an arbitrary superedge, say $\{u, v\}$, and the associated outer edge from (u, x, k_L) to (v, y, k_R) (where here x and y are the same as in our special bucket). It is clear that by construction the only paths of length at most k which can span an outer edge are either the outer edge itself or the canonical paths. In the former case we explicitly added an arbitrary pair of nodes that cover $\{u, v\}$. In the second case, the existence of a canonical path in the spanner means that the connection edges it uses are in the bucket. This in turn means that the inner nodes they are incident on were added to the REP-cover, and since the canonical path uses the inner edge between them they must in fact cover the $\{u, v\}$ superedge. Thus we have a valid REP-cover of size less than $s(|U| + |V|)$. ◀

We can now use Lemmas 14 and 15 to prove the desired hardness for Directed LDkS.

▶ **Theorem 16.** *Unless* $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{\mathrm{polylog}(n)})$*, there is a constant* $\gamma > 0$ *so that no polynomial time algorithm can approximate Directed LDkS to a factor better than* Δ^γ *(for any integer* $k \geq 3$*).*

Proof. Lemmas 14 and 15, when combined with Theorem 13, imply hardness of $\Omega(s)$. With the chosen value of d_H, it is easy to verify that Δ is achieved at either maximal or minimal outer nodes. The degree in G of the former is at most $d_{G'}d_H + 1 = O(d^2_{G'} + d_{G'}|\Sigma|)$, while the latter have degree at most $|\Sigma|d_H + 1 = O(d_{G'}|\Sigma| + |\Sigma|^2)$. If $k = 3$ or 4 then the the maximal nodes might also be minimal, and so have degree equal to the sum of those bounds. But for any $k \geq 3$ we have that $\Delta \leq O(d^2_{G'} + |\Sigma|^2)$. Since we specifically chose to use hard Min-Rep instances where $d_{G'}$ and $|\Sigma|$ are polynomial in s, this proves the theorem. ◀

3.2 Undirected LDkS

We now want to handle the undirected case (again, we only give an outline). This is complicated primarily because switching edges to being undirected creates new paths that the spanner might use. In the directed setting, if an outer edge was not in the spanner then the only way for it to be spanned was to use a canonical path, which essentially determined the "suggested" REP-cover for the Min-Rep instance. Once we move to the undirected setting there is another possibility: an outer edge could be spanned by a path consisting entirely of outer edges. This was not possible with directed edges because all outer edges were directed into V_{out}^R. These new paths are problematic, since if an outer edge is spanned in this way there is no suggested REP-cover. Thus we will try to make sure that no such paths actually exist.

We will need to start with hard Min-Rep instances with some extra properties: namely, we want large supergirth and $d_{G'} \geq |\Sigma|$. This can be achieved using a simple modification of [11], giving the following lemma.

▶ **Lemma 17.** *Unless* $\mathrm{NP} \subseteq \mathrm{BPTIME}(n^{\mathrm{polylog}(n)})$*, there is no polynomial time algorithm that can distinguish between instances of Min-Rep in which there is a REP-cover of size* $|U| + |V|$ *(i.e. a YES instance) and instances in which every REP-cover has size at least* $s(|U| + |V|)$*, even when all instances are guaranteed to have the following properties:*
1. *The girth of the supergraph is larger than* $k + 1$*,*
2. *There is some value* $d_{G'}$ *so that all degrees in the supergraph are within a factor of* 2 *of* $d_{G'}$*,*
3. $s, d_{G'}$*, and* $|\Sigma|$ *are all polynomials of each other, and*
4. $d_{G'} \geq |\Sigma|$*.*

We will also use a balanced regular bipartite graph H as before, but instead of being the (directed) complete bipartite graph, H will be a balanced regular bipartite graph of girth at least $k + 2$ and degree d_H (note that such graphs exist as long as the number of nodes $n_H = |X| + |Y| = 2|X|$ is sufficiently large, e.g. as long as $n_H d_H \leq n_H^{1+\frac{1}{3k^2}}$ [3]). We will set $d_H = d_{G'}$, so the number of outer edges incident on each maximal outer node of G is $d = d_H d_{G'} = d_{G'}^2$.

We start with the same graph G as in the directed setting (although with undirected edges, and using Min-Rep instances from Lemma 17).

We will then subsample in essentially the same way as [11]: for every outer edge $\{(u, x, k_L), (v, y, k_R)\}$ we will flip an independent coin, keeping the edge with probability $p = \frac{\alpha \log |\Sigma|}{d}$ and removing it with probability $1 - p$ (we will set $\alpha = d^{\frac{k+2}{2(k+1)}}/(4 \log |\Sigma|)$). If we remove it we will also remove the associated inner edges, i.e. we will remove all inner edges of the form $\{(a, \{x, y\}), (b, \{x, y\})\}$ where $a \in \Gamma(u)$ and $b \in \Gamma(v)$. This gives us a new graph G_α.

Call an outer edge of G_α *bad* if it is part of a cycle in G_α consisting only of outer edges of length at most $k + 1$. We will see that there are not too many bad edges, so we then create our final instance of LDkS by removing all bad edges (and associated inner edges) from G_α, giving us a new graph $\widehat{G_\alpha}$. Intuitively $\widehat{G_\alpha}$ is essentially the same as G_α, since there are so few bad edges in G_α.

3.2.1 Analysis

We can still build a spanner using canonical paths corresponding to a REP-cover of each subsampled instance, so if we start with a YES instance we can still build a spanner of $\widehat{G_\alpha}$ with small maximum degree. This is essentially the same as Lemma 14.

▶ **Lemma 18.** *If we started out with a YES instance of Min-Rep, there is a k-spanner of* $\widehat{G_\alpha}$ *with maximum degree at most* $\max\{d_H + 1, d_{G'}^{\frac{1}{k+1}} + 2|\Sigma| + 1\}$*.*

For each outer edge $((u, x, k_L), (v, y, k_R))$ in G, call a path from (u, x, k_L) to (v, y, k_R) *bad* if it contains only outer edges and has length at most k (and larger than 1). So an outer edge is bad if and only if there is a bad path between its endpoints. We begin by analyzing the number of bad paths for any fixed outer edge in the original construction G (before subsampling). The trivial bound would be d^{k-1}, but because G' and H both have large girth we can do better. This is the reason we needed to start out with already subsampled instances of Min-Rep (i.e. why we had to start with instances based on [11] rather than generic hard Min-Rep instances, like those from Theorem 13).

▶ **Lemma 19.** *For any outer edge, the number of bad paths is at most* $O(4^k d^{\frac{k-1}{2}})$.

Lemma 19 now allows us to upper bound the number of bad edges in G_α, since we set α to be low enough that we expect all of the bad paths in G to be missing at least one edge in G_α.

▶ **Lemma 20.** *With probability at least* $3/4$ *the number of outer edges in* G_α *that are bad is at most* $|U| \cdot |X| \cdot d_H$

Recall that our construction started with $|E_H| = |X|d_H$ copies of the original Min-Rep instance, and each outer edge is associated with a single such instance. So in G_α the average instance has at most $|U|$ bad edges, and thus by Markov at least $|E_H|/2$ of the instances have at most $2|U|$ bad edges. It is well-known that removing only $|U|$ superedges of a Min-Rep instance affects the size of the optimal REP-cover in a NO instance by at most a constant factor (see e.g. [11] for a proof of this). So $\widehat{G_\alpha}$ is essentially G_α. So now that there are no bad edges, the only ways to span an outer edge in $\widehat{G}_\alpha$ are the edge itself or a canonical path, so we are essentially back to the directed case (except that we can only use 1/2 of the $|E_H|$ Min-Rep instances to prove our bound, but that is plenty). This implies that in a NO instance all spanners must have large maximum degree, through an analysis similar to Lemma 15.

▶ **Lemma 21.** *It we started out with a NO instance of Min-Rep, any k-spanner of* $\widehat{G_\alpha}$ *must have a node of degree at least* $\tilde{\Omega}(d_h \cdot d_{G'}^{\frac{1}{2(k+1)}})$.

The main hardness theorem is now implied by the chosen parameters.

▶ **Theorem 22.** *Unless* $\mathrm{NP} \subseteq \mathrm{BPTIME}(n^{\mathrm{polylog}(n)})$*, there is no algorithm that can approximate* Lowest Degree k-Spanner *on undirected graphs to a factor better than* $\Delta^{\Omega(1/k)}$ *(for any integer* $k \geq 3$*).*

References

1 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.

2 Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. Additive spanners and (α, β)-spanners. *ACM Trans. Algorithms*, 7(1):5:1–5:26, December 2010.

3 Mohsen Bayati, Andrea Montanari, and Amin Saberi. Generating random graphs with large girth. In *SODA'09*, pages 566–575, 2009.

4 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Improved approximation for the directed spanner problem. In *ICALP (1)*, pages 1–12, 2011.

5 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. In *SODA'09*, pages 932–941, 2009.

6 T.-H. Hubert Chan, Michael Dinitz, and Anupam Gupta. Spanners with slack. In *Proceedings of the 14th Annual European Symposium on Algorithms*, ESA, pages 196–207, 2006.

7 Barun Chandra, Gautam Das, Giri Narasimhan, and José Soares. New sparseness results on graph spanners. *International Journal of Computational Geometry and Applications*, 5(1):125–144, 1995.

8 S. Chechik, M. Langberg, David Peleg, and L. Roditty. Fault-tolerant spanners for general graphs. In *STOC'09*, pages 435–444, New York, NY, USA, 2009. ACM.

9 Shiri Chechik. New additive spanners. In *SODA'13*, pages 498–512, 2013.

10 Eden Chlamtáč, Michael Dinitz, and Robert Krauthgamer. Everywhere-sparse spanners via dense subgraphs. *FOCS'12*, 0:758–767, 2012.

11 Michael Dinitz, Guy Kortsarz, and Ran Raz. Label cover instances with large girth and the hardness of approximating basic k-spanner. In *ICALP'12*, pages 290–301, Berlin, Heidelberg, 2012. Springer-Verlag.

12 Michael Dinitz and Robert Krauthgamer. Directed spanners via flow-based linear programs. In *STOC'11*, pages 323–332, New York, NY, USA, 2011. ACM.

13 Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: Better and simpler. In *PODC'11*, pages 169–178, 2011.

14 Yevgeniy Dodis and Sanjeev Khanna. Designing networks with bounded pairwise distance. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, STOC'99, pages 750–759, New York, NY, USA, 1999. ACM.

15 Michael Elkin and David Peleg. The hardness of approximating spanner problems. In *STACS*, pages 370–381, 2000.

16 Michael Elkin and David Peleg. Strong inapproximability of the basic k-spanner problem. In *ICALP*, pages 636–647, 2000.

17 Michael Elkin and David Peleg. Approximating k-spanner problems for $k > 2$. *Theor. Comput. Sci.*, 337(1-3):249–277, 2005.

18 Michael Elkin and Shay Solomon. Fast constructions of light-weight spanners for general graphs. In *In Proc. of 24th SODA*, pages 513–525, 2013.

19 Guy Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

20 Guy Kortsarz and David Peleg. Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994.

21 Guy Kortsarz and David Peleg. Generating low-degree 2-spanners. *SIAM J. Comput.*, 27(5):1438–1456, 1998.

22 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

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

24 Mohit Singh and Lap Chi Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *STOC'07*, pages 661–670, 2007.

25 Mikkel Thorup and Uri Zwick. Compact routing schemes. In *SPAA'01*, pages 1–10, New York, NY, USA, 2001. ACM.

26 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, January 2005.