THEORY OF COMPUTING, Volume 12 (15), 2016, pp. 1–29
www.theoryofcomputing.org

SPECIAL ISSUE: APPROX-RANDOM 2014

Lowest-Degree k-Spanner: Approximation and Hardness

Eden Chlamtáč* Michael Dinitz†

Received April 7, 2015; Revised July 20, 2016; Published September 24, 2016

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 $\widetilde{O}(\Delta^{3-2\sqrt{2}})$-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 arbitrary constant k. Specifically, we give an LP-based $\widetilde{O}(\Delta^{(1-1/k)^2})$-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.

ACM Classification: G.2.2, F.2.2, G.1.6

AMS Classification: 68W25, 68Q17, 05C85

Key words and phrases: approximation algorithms, hardness of approximation, graph spanners

An extended abstract of this paper appeared in the Proceedings of the 17th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2014) [12].

*Partially supported by ISF grant 1002/14.

†Partially supported by NSF grants 1464239 and 1535887.

 DOI: 10.4086/toc.2016.v012a015

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[1] in H and G, respectively.[2] Graph spanners were originally introduced in the context of distributed computing [25, 26], 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 [9]). Spanners have also appeared as fundamental building blocks in a wide range of applications, from routing in computer networks [29] to property testing of functions [6].

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 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 [28] 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, we define LDkS to be the problem of minimizing the maximum total degree of a k-spanner, that is, the sum of the in- and out-degrees. Kortsarz and Peleg initiated the study of the maximum degree of a spanner, giving an $O(\Delta^{1/4})$-approximation for LD2S [24] (where Δ is the maximum degree of the input graph). This was only recently improved to $O(\Delta^{3-2\sqrt{2}+\varepsilon}) = O(\Delta^{0.17...+\varepsilon})$ for arbitrarily small constant $\varepsilon > 0$ by Chlamtáč, Dinitz, and Krauthgamer [13]. The only known hardness for LD2S was $\Omega(\log n)$ [24]. 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 constant $k \geq 3$. 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.

[1] We consider both undirected and directed graphs. In directed graphs, "path" refers to directed path (hence the distance function $d_G(u,v)$ is not symmetric).

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

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 1.1. *For every fixed integer $k \geq 1$, there is a randomized polynomial-time $\widetilde{O}(\Delta^{(1-1/k)^2})$-approximation for* LOWEST-DEGREE k-SPANNER *in both undirected and directed graphs.*

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 [24], although not the improved one of [13]. This is not a coincidence: our algorithm is a modification of [24], 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 [15, 6] but with a degree objective, and reduces to the LP used by [24] when $k = 2$).

The LP rounding in [24] 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 $u \rightsquigarrow v$ path survives the rounding with probability at least $1 - 1/\mathrm{e}$.

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 $1/\operatorname{polylog}(\Delta)$ 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 1.2. *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 $\mathsf{NP} \subseteq \mathsf{BPTIME}(2^{\operatorname{polylog}(n)})$.*

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

Theorem 1.3. *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 $\mathsf{NP} \subseteq \mathsf{BPTIME}(2^{\operatorname{polylog}(n)})$.*

[3] Our algorithm and analysis work for both the undirected and directed cases with no change. The parameter k is taken to be a constant, and the $\widetilde{O}$ notation in the approximation guarantee hides polylogarithmic factors of the form $O(\log n(\log\Delta)^{f(k)})$ for some function $f(k)$ that depends exponentially on k. The running time is bounded by a fixed polynomial in n that does not depend on k.

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 [22] and Elkin and Peleg [20]) 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 [18] 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 [14] 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 [18]). So we might instead hope that we could simply use the *ideas* of [14], 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 [14], 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 [25, 26] 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 [4, 10], fault-tolerance [11, 16], or average stretch [8]) or considering algorithmic aspects such as allowing fast distance queries [30] or extremely fast constructions [21].

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 [23] and then an $O(\Delta^{1/4})$-approximation for LOWEST-DEGREE 2-SPANNER [24]. This was followed by upper bounds by Elkin and Peleg [19] for a variety of related spanner problems including LD2S (although not LDkS).

With the exception of [24], 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 [24], 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 [7], directed spanners [15, 6], fault-tolerant spanners [15, 16], and LD2S [13]. In this paper we use a rounding scheme similar to [24] (with a much more complicated analysis) and an LP that is a degree-based variant of the flow-based LP introduced by [15] (an earlier use of flow-based LPs for approximating spanners is [17]).

On the hardness side, the first results were due to Kortsarz [22] who proved $\Omega(\log n)$-hardness for the basic k-spanner problem (for constant k) and $2^{\log^{1-\varepsilon} n}$-hardness for a weighted version. These results were pushed further by Elkin and Peleg [20], who proved the same $2^{\log^{1-\varepsilon} n}$-hardness for a collection of spanner problems including directed k-spanner. Separately, Kortsarz and Peleg proved logarithmic hardness for LD2S [24]. 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 [14]. 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 (in the case of directed graphs, a directed path). We think of paths as tuples of edges, and denote by $(p)_i$ the i-th edge in a path p. For integer k, we will use $[k]$ to denote the set $\{1,2,\dots,k\}$. We will use $\ln x$ to denote $\log_e x$, and $\log x$ to denote $\log_2 x$.

A *k-spanner* of G is a subgraph H of G in which $d_H(u,v) \le k \cdot d_G(u,v)$ 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 1.4. 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 1.1. 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 $\sum_{i=1}^{k} D^i = O(D^k)$ (the maximum number of possible paths of length $\le k$ starting from a given node in the spanner). Therefore, we have

Observation 2.1. 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.

$$\begin{array}{lll} \min & d & \\ \text{s.t.} & \sum_{e\sim v} x_e \le d & \forall v \in V \quad (2.1) \\ & \sum_{p:u\rightsquigarrow v, |p|\le k} y_p = 1 & \forall (u,v) \in E \quad (2.2) \\ & x_e \ge \sum_{\substack{p:u\rightsquigarrow v,|p|\le k \\ p\ni e}} y_p & \forall (u,v), e \in E \quad (2.3) \\ & x_e, y_p \ge 0 & \forall e, p \quad (2.4) \end{array}$$

Note that this LP has size $n^{k+O(1)}$, and thus can be solved in polynomial time when k is constant. However, there is also an equivalent formulation whose size is bounded by a fixed polynomial in n, that does not depend on k. As shown in [6], when the edges have unit length (as in our case), constraints (2.2) and (2.3) can be captured by general flow constraints on a layered graph with $(k+1)n$ vertices. Since we may assume that $k \le n-1$ (in the unit-length case, if $k > n-1$, then a subgraph is a k-spanner iff it is an $(n-1)$-spanner), this graph has at most n^2 vertices, which allows us to solve the LP in time which is a fixed polynomial in n.

Recall that in a k-spanner, it is sufficient to span every edge by a path of length at most 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, we get the following easy observation.

Observation 2.2. 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 d_{LP}.

We apply a rather naïve rounding algorithm to the LP solution, which can be thought of as a natural extension of the rounding in [24] 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 $\widetilde{\Omega}(1)$. It is then only a matter of repeating the above algorithm a polylogarithmic number of times to ensure that every edge is spanned with high probability. 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 2.3. *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 $O(d_{\text{OPT}}\Delta^{(1-1/k)^2})$.*

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

$$\begin{aligned}
\textstyle\sum_{e\sim v} x_e^{1/k} &\le (\deg_G(v))^{1-1/k} \left(\textstyle\sum_{e\sim v} x_e\right)^{1/k} && \text{by Jensen's inequality}^4 \\
&\le \Delta^{1-1/k} d_{\mathrm{LP}}^{1/k} \\
&\le \Delta^{1-1/k} d_{\mathrm{LP}}^{1/k} \cdot \frac{d_{\mathrm{OPT}}^{1-1/k}}{\Delta^{(1/k)(1-1/k)}} && \text{by Observation 2.1} \\
&\le \Delta^{(1-1/k)^2} \cdot d_{\mathrm{OPT}} && \text{by Observation 2.2.}
\end{aligned}$$

□

Remark 2.4. 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. Specifically, for any vertex for which the expected degree is $\mu \ge 2\ln n$, since the degree is a sum of independent Bernoulli variables, the probability that its degree will be greater than 3μ is (by Chernoff) at most $(\mathrm{e}^2/(3^3))^{\mu} < \mathrm{e}^{-\mu} = 1/n^2$. On the other hand, if the expected degree is less than $2\ln n$, then the probability that the degree will be greater than $6\ln n$ is even smaller. Taking a union bound, the probability that any vertex v with expected degree μ_v will have degree greater than $\max\{3\mu_v, 6\ln n\}$ is at most $1/n$. This concentration argument can also be applied to the total number of edges incident to a vertex added throughout the polylogarithmically many iterations of the basic algorithm, with multiplicities.

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

2.2 Proof of correctness

Having shown that t iterations of the algorithm (for $t \ge 3\ln n$) produce a subgraph with maximum degree at most $O(t\Delta^{(1-1/k)^2} d_{\mathrm{OPT}})$, it suffices to show that, for every edge, one iteration of the algorithm spans that edge with probability at least $\Omega(1/\operatorname{polylog}(\Delta))$. Theorem 1.1 then follows by choosing some $t = O(\log n \cdot \operatorname{polylog}(\Delta))$. Thus, our goal for the remainder of the section is to prove the following lemma.

Theorem 2.5. *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* $1/\operatorname{polylog}(\Delta)$.

Before proving this theorem, let us give some intuition. Suppose, for simplicity, that for an edge $(u,v) \in E$, all the contribution in (2.2) (the spanning constraint) comes from paths of length exactly k. First, consider the case where all the paths with non-zero weight y_p in (2.2) are edge-disjoint. For every edge e in such a path p we have, from (2.3), 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

$$\prod_{e\in p} x_e^{1/k} \ge y_p\,.$$

[4] Alternatively, this follows from Hölder's inequality.

Denoting by P the set of such paths, by disjointness these events are independent, and therefore we have

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

Thus, repeating this process $O(\log n)$ times, all such edges will be spanned with high probability.

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 $(e_i)_{i=1}^k \in \prod_{i=1}^k E_i$ for some disjoint edge sets $E_1,\ldots,E_k \subset E$ (see Lemma 2.6). Consider the other extreme (in terms of path disjointness) where $k' = k$ and the flow is distributed evenly over all possible paths of the form $u - v_1 - v_2 - \cdots - v_{k-1} - v$ for $v_i \in V_i$, where $\{V_i \mid i \in [k-1]\}$ is an equipartition of $V \setminus \{u,v\}$ (that is, uniform flow through a series of bicliques). 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 $\approx (n/(k-1))^{1-2/k}$ 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 $\approx (n/(k-1))^{(k^2-3k+2)/k}$ 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 with high probability due to concentration). However, this is not easy to formalize (moreover, we note that obvious analysis techniques do not work—for example, 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 P' of paths and a (small) set S of edges, 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, which we prove in Section 2.3.

Lemma 2.6. *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 $\sum_{p\in P} y_p \geq 1/\mathrm{polylog}(\Delta)$, there exists a subset $P' \subseteq P$ satisfying the following conditions.*

- *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 $e_j \in (e_i)_{i\in I}$ then $m_{P'}((e_i)_{i\in I}) \le m(e_j)$ and therefore $m_I \le m_{\{j\}}(\log\Delta)^{f(k)}$ for $j \in I$.)*

We note that if $\prod_{i=1}^{k'} m_{\{i\}} \le \operatorname{polylog}(\Delta)$, 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 2.7. *Let P' be the set of paths given by Lemma 2.6, and suppose $\prod_{i=1}^{k'} m_{\{i\}} < (\log\Delta)^g$ for some constant $g = g(k)$. Then with probability at least $1/(\log\Delta)^h$ (for some constant $h = h(k)$), 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 $i = 1,\dots,k'$ 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 $m_{P'}(e) \le m_{\{i\}}(\log\Delta)^{f(k)}$, 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 $|P^*| \ge |P'|/(\log\Delta)^{g(k)+k'f(k)}$, and the paths in P^* are edge-disjoint.

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

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

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

$$\begin{aligned}
1 - \prod_{p\in P^*}\Big(1 - \prod_{e\in p} x_e^{1/k}\Big) &\ge 1 - \Big(1 - \big(|P'|(\log\Delta)^{f(k)}\big)^{-k'/k}\Big)^{|P^*|} \\
&\ge 1 - \exp\Big(-|P'|^{(k-k')/k}(\log\Delta)^{-(f(k)k'/k+g(k)+k'f(k))}\Big) \\
&\ge 1 - \exp\Big(-(\log\Delta)^{-(1+1/k)(f(k)+g(k))}\Big) \\
&= (1-o(1))(\log\Delta)^{-(1+1/k)(f(k)+g(k))}.
\end{aligned}$$

□

We can also easily deal with the case $m_{\{i\}} \ge |P'|/\operatorname{polylog}(\Delta)$, 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 2.9). 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.

Before we describe this part of the analysis, let us describe a more uniform rounding scheme, which is dominated by our algorithm. Consider a single edge $e \in E_i$. We know this edge is contained in $m_{P'}(\{e\})$ paths in P', and each of these paths has weight $y_p \in [y_0, 2y_0]$. Therefore, by (2.3) and Lemma 2.6, we have

$$x_e \ge m(\{e\})y_0 \ge m_{\{i\}}y_0 \ge m_{\{i\}}/(|P'|(\log\Delta)^{f(k)}). \tag{2.5}$$

Suppose instead of sampling each edge independently with probability $x_e^{1/k}$, we retained every 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

$$\mathbb{E}[Y] = |P'|^{1-k'/k}\left(\prod_{i=1}^{k'} m_{\{i\}}\right)^{1/k} / (\log\Delta)^{f(k)k'/k}, \tag{2.6}$$

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

$$\mathrm{Prob}[Y=0] \le \mathrm{Prob}\left[Y < \frac{1}{2}\mathbb{E}[Y]\right] \le \mathrm{Prob}\left[(Y-\mathbb{E}[Y])^2 > \frac{1}{4}(\mathbb{E}[Y])^2\right] < \frac{4\,\mathrm{Var}[Y]}{(\mathbb{E}[Y])^2}.$$

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

$$\mathrm{Var}[Y] = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 < \frac{1}{8}(\mathbb{E}[Y])^2. \tag{2.7}$$

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.

Lemma 2.8. *Let P' be the set of paths given by Lemma 2.6. Then if*

$$\prod_{i=1}^{k'} m_{\{i\}} \ge (\log\Delta)^{g(k)},$$

and for every $i \in [k']$ we have $m_{\{i\}} \le |P'|(\log\Delta)^{-g(k)}$, where $g(k) \ge (4k+2)f(k)$ then (2.7) holds.

Proof. Recall that Y is defined w. r. t. the modified rounding (in which edges in E_i are all chosen with probability $x_i^{1/k}$). For any tuple of edges (e_i), we introduce the following random variable:

$$X_{(e_i)} = \begin{cases} 1 & \text{all edges in } (e_i) \text{ survive the modified rounding,} \\ 0 & \text{otherwise.} \end{cases}$$

Note that, in particular, we have

$$Y = \sum_{p \in P'} X_p.$$

For every $I \subseteq [k']$, we also denote by E_I the set of all tuples $(e_i) \in \prod_{i\in I} E_i$ that are contained in at least one path in P' (note that $E_\emptyset = \{()\}$ and $E_{[k']} = P'$). Note that $|E_I| m_I \le |P'|$. We can now bound the second moment as follows.

$$
\begin{aligned}
\mathbb{E}[Y^2] = \mathbb{E}\Big[\big(\sum_{p\in P'} X_p\big)^2\Big] &= \sum_{p_1\in P'}\sum_{p_2\in P'} \mathbb{E}[X_{p_1}X_{p_2}] \\
&= \sum_{I\subseteq[k']}\sum_{(e_i)\in E_I}\sum_{\substack{p_1,p_2\in P'\\ p_1\cap p_2=(e_i)}} \mathbb{E}[X_{p_1}X_{p_2}] \\
&= \sum_{I\subseteq[k']}\sum_{(e_i)\in E_I} \mathbb{E}[X_{(e_i)}] \sum_{\substack{p_1,p_2\in P'\\ p_1\cap p_2=(e_i)}} \mathbb{E}[X_{p_1\setminus(e_i)}]\,\mathbb{E}[X_{p_2\setminus(e_i)}] \\
&< \sum_{I\subseteq[k']}\sum_{(e_i)\in E_I} \mathbb{E}[X_{(e_i)}] \left(\sum_{p:((p)_i)_{i\in I}=(e_i)} \mathbb{E}[X_{p\setminus(e_i)}]\right)^2 .
\end{aligned}
$$

Noting that in the final sum, for $I=\emptyset$ the summand equals $(\sum_{p\in P'}\mathbb{E}[X_p])^2$, we get

$$
\mathbb{E}\left[\big(\sum_{p\in P'} X_p\big)^2\right] - \left(\mathbb{E}\,\big[\sum_{p\in P'} X_p\big]\right)^2 < \sum_{\emptyset\neq I\subseteq[k']}\sum_{(e_i)\in E_I} \mathbb{E}[X_{(e_i)}]\left(\sum_{p:((p)_i)_{i\in I}=(e_i)} \mathbb{E}[X_{p\setminus(e_i)}]\right)^2 . \tag{2.8}
$$

Thus, we only have to bound the sum on the right hand side above. We treat the summand for $I=[k']$ separately. Using our lower bound on $\prod_i m_i$, for this value of I we get

$$
\begin{aligned}
\sum_{p\in P'}\mathbb{E}[X_p]\left(\sum_{p'=p}\mathbb{E}[X_\emptyset]\right)^2 &= \sum_{p\in P'}\mathbb{E}[X_p] \\
&= \left(\sum_{p\in P'}\mathbb{E}[X_p]\right)^2 (\log\Delta)^{f(k)k'/k}\prod_{i=1}^{k'} m_{\{i\}}^{-1/k}|P|^{-(1-k'/k)} \qquad \text{by (2.6)} \\
&\le (\log\Delta)^{-(g(k)-f(k)k')/k}\left(\sum_{p\in P'}\mathbb{E}[X_p]\right)^2 . \qquad (2.9)
\end{aligned}
$$

Finally, for all non-empty strict subsets $\emptyset \neq I \subsetneq [k']$ we have the following.

$$\begin{aligned}
\sum_{(e_i)\in E_I} \mathbb{E}[X_{(e_i)}] \left(\sum_{p:((p)_i)_{i\in I}=(e_i)} \mathbb{E}[X_{p\setminus(e_i)}] \right)^2 &= \sum_{(e_i)\in E_I} \prod_{i\in I} x_i^{1/k} \left(m_{P'}(e_i) \prod_{j\in[k']\setminus I} x_j^{1/k} \right)^2 \\
&\le (\log\Delta)^{2f(k)} |E_I| m_I^2 \prod_{i\in I} x_i^{1/k} \prod_{j\in[k']\setminus I} x_j^{2/k} \\
&\le (\log\Delta)^{2f(k)} |P'| m_I \prod_{i\in I} x_i^{1/k} \prod_{j\in[k']\setminus I} x_j^{2/k} \\
&= (\log\Delta)^{2f(k)} \left(\mathbb{E}\big[\sum_{p\in P'} X_p \big] \right)^2 |P'|^{-1} m_I \prod_{i\in I} x_i^{-1/k} \\
&= (\log\Delta)^{2f(k)} \left(\sum_p \mathbb{E}[X_p] \right)^2 \prod_{i\in I} \frac{m_I^{1/|I|}}{x_i^{1/k} |P'|^{1/|I|}} \\
&= (\log\Delta)^{(2+|I|/k)f(k)} \left(\sum_p \mathbb{E}[X_p] \right)^2 \prod_{i\in I} \frac{m_I^{1/|I|}}{m_{\{i\}}^{1/k} |P'|^{1/|I|-1/k}} \\
&\le (\log\Delta)^{(3+|I|/k)f(k)} \left(\sum_p \mathbb{E}[X_p] \right)^2 \prod_{i\in I} \frac{m_{\{i\}}^{1/|I|-1/k}}{|P'|^{1/|I|-1/k}} .
\end{aligned}$$

By our upper bound on $m_{\{i\}}$, this gives

$$\sum_{(e_i)\in E_I} \mathbb{E}[X_{(e_i)}] \left(\sum_{p:((p)_i)_{i\in I}=(e_i)} \mathbb{E}[X_{p\setminus(e_i)}] \right)^2 \le \left(\sum_p \mathbb{E}[X_p] \right)^2 (\log\Delta)^{-(1-|I|/k)g(k)+(3+|I|/k)f(k)} ,$$

and combining this with (2.9) and plugging into (2.8), we get (by our choice of $g(k)$)

$$\begin{aligned}
\mathbb{E}\left[\big(\sum_{p\in P'} X_p \big)^2 \right] - \left(\mathbb{E}\big[\sum_{p\in P'} X_p \big] \right)^2 &< \left(\sum_{p\in P'} \mathbb{E}\left[X_p\right] \right)^2 (\log\Delta)^{-(g(k)-f(k)k')/k} \\
&\quad + \left(\sum_{p\in P'} \mathbb{E}\left[X_p\right] \right)^2 \cdot \sum_{i=1}^{k'-1} \binom{k'}{i} (\log\Delta)^{-(1-i/k)g(k)+(3+i/k)f(k)} \\
&< \left(\sum_{p\in P'} \mathbb{E}\left[X_p\right] \right)^2 2^k (\log\Delta)^{-f(k)/k} ,
\end{aligned}$$

and the lemma follows for all sufficiently large Δ. □

Finally, we combine these three components to give our correctness guarantee. Theorem 2.5 will follow by applying Lemma 2.6 applied to the set P of all $u \rightsquigarrow v$ paths of length at most k, and then applying the following theorem to the set P' of paths guaranteed by the lemma.

Theorem 2.9. *Let P' be a collection of $u \rightsquigarrow v$ paths as in Lemma 2.6, then with probability at least $1/(\log\Delta)^{\ell(k)}$ (for some function ℓ), at least one path in P' survives the rounding.*

Proof. First, consider the case where $m_{\{i\}} \le |P'|(\log\Delta)^{-(4k+2)f(k)}$ for every $i \in [k']$. In this case, if

$$\prod_{i=1}^{k'} m_{\{i\}} \ge (\log\Delta)^{(4k+2)f(k)},$$

then the theorem follows directly from our second moment argument and Lemma 2.8. If

$$\prod_{i=1}^{k'} m_{\{i\}} < (\log\Delta)^{(4k+2)f(k)},$$

on the other hand, then the theorem follows from Lemma 2.7.

On the other hand, if there does exist some $i \in [k']$ such that $m_{\{i\}} \ge |P'|(\log\Delta)^{-(4k+2)f(k)}$, then the above analysis breaks down. In this case, choose any edge $e \in E_i$, and note that (by (2.5))

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

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 2.6 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 $(\log\Delta)^{-\ell(k-1)}$ 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 $(\log\Delta)^{-(4+3/k)f(k)-\ell(k-1)}$. □

2.3 Bucketing and pruning

Here we prove Lemma 2.6. Consider an edge $(u,v) \in E$. It is immediate that for some $k' \in [k]$ at least $1/k$ fraction of the flow in P must come from paths of length exactly k', which guarantees the first condition of the lemma. The third condition follows by noting that we can easily bucket by path weight (flow) and lose at most a logarithmic factor.

Claim 2.10. *For any set P of paths of length k', there is some positive integer $t < k'\log\Delta$ such that the set $P_t = \{p \in P \mid 2^{-(t+1)} < y_p \le 2^{-t}\}$ of paths satisfies*

$$\sum_{p \in P_t} y_p \ge \frac{1}{k'\log\Delta}\Big(\sum_{p\in P} y_p - \frac{1}{\Delta}\Big).$$

The second condition requires the paths to form a layer graph, that is, for every edge $e \in E$ there is a fixed $i \in [k]$ such that edge e can only appear as the i-th edge in a $u \rightsquigarrow v$ flow-path. This can be done while losing at most a constant factor (that depends on k), as shown in the following lemma.

Lemma 2.11. *Given P, a collection of $u \rightsquigarrow v$ paths of length k, with non-negative weights y_p, there is a subset $P' \subseteq P$ and a mapping $\varphi : E \to [k]$ such that every edge $e \in E$ can only appear in the $\varphi(e)$-th position in any path in P', and furthermore*

$$\sum_{p\in P'} y_p \geq (k-2)^{-(k-2)} \sum_{p\in P} y_p \,.$$

Proof. We may assume that all paths in P are simple paths. Note that any edge incident to u or to v will always be the first or k-th edge, respectively. Thus, for $k \leq 2$, the lemma is trivially true for $P' = P$. Now consider $k \geq 3$, and for all other edges, choose a uniformly random assignment $\varphi : E \setminus (E(u) \cup E(v)) \to \{2, \ldots, k-1\}$. Since the paths are simple, the probability that any path $p \in P$ is consistent with φ (that is, φ maps the edge $(p)_i$ to i, for every $i \in [k]$) is exactly $(k-2)^{-(k-2)}$. Let P' be the set of paths in P that are consistent with φ, then by linearity of expectation we have

$$\mathbb{E}\Big[\sum_{p\in P'} y_p\Big] = (k-2)^{-(k-2)} \sum_{p\in P} y_p \,,$$

and so there exists a set $P' \subseteq P$ with the required properties. $\square$

We now have a layered collection P of paths of length exactly k' with nearly uniform (up to a constant factor) path weights which support $\widetilde{\Omega}(1)$ flow from u to v (where the constant depends on k). Let $E_1, \ldots, E_{k'} \subseteq E$ be the layer sets corresponding to paths P. We would like to further prune the set P so that the combinatorial structure of the flow network is almost regular in the sense required by the final condition in Lemma 2.6.

Consider any subset $I \subseteq [k']$, and for every tuple $(e_i)_{i\in I} \in \prod_{i\in I} E_i$, denote by $m_P((e_i)_{i\in I})$ the number of paths in P which use these edges. That is,

$$m_P((e_i)_{i\in I}) = |\{p \in P \mid (|p| = k') \wedge (\forall i \in I : (p)_i = e_i)\}| \,.$$

The proof of Lemma 2.6 is complete by showing that the paths in P can now be pruned so that the values $m((e_i)_{i\in I})$ are roughly uniform (for all tuples corresponding to a given $I \subset [k']$), and a polylogarithmic fraction of paths is retained.

Lemma 2.12. *There exists some function $f : \mathbb{N} \to \mathbb{N}$ such that, given a layered set P of paths of length k', we can find a subset $P' \subseteq P$ satisfying*

- *There exists a vector $(m_I)_{I\subseteq[k']}$ so that for every $I \subseteq [k']$ and every edge tuple $(e_i)_{i\in I} \in \prod_{i\in I}$ we have either $m_{P'}((e_i)_i) = 0$ or $m_{P'}((e_i)_i) \in [m_I/(\log\Delta)^{f(k')}, m_I]$.*
- $|P'| \geq |P|/(\log\Delta)^{f(k')}$.

Proof. Start by letting $P' = P$, the set of all paths. Sequentially, for each $\emptyset \neq I \subseteq [k;]$, do the following: bucket all possible tuples $(e_i)_{i\in I} \in \prod_{i\in I} E_i$ according to their current $m_{P'}((e_i)_i)$ values. Noting that this value is at most $\Delta^{k'-1}$ (the maximum possible number of $u \rightsquigarrow v$ paths of length k'), there is some m_I such that tuples with $m((e_i)_i \in [m_I/2, m_I]$ constitute at least a $1/((k'-1)\log\Delta)$ fraction of paths in P'. Retain paths corresponding to these tuples and delete all other paths from P'.

Note that since this procedure is repeated a constant $(2^{k'}-1)$ number of times, we still have

$$|P'| \geq |P|/((k'-1)\log\Delta)^{2^{k'}-1}.$$

Now consider some set $I \subseteq [k]$. Immediately after the pruning step corresponding to I, all remaining I-tuples had $m((e_i)_i) \in [m_I/2, m_I]$. Denote the set of these tuples by T_I, and the set of paths that survive this step by P'_I. We have that the average $m(\cdot)$ value for I-tuples at this stage is

$$\mathbb{E}_{(e_i)\in T_I}[m_{P'_I}((e_i)_i)] = |P'_I|/|T_I| \in [m_I/2, m_I].$$

After further pruning (at most $(2^{k'}-2)$ steps for other subsets $I' \subseteq [k]$), this regularity no longer holds. However, since a polylogarithmic fraction of paths are retained, the average $m((e_i)_i)$ value for $(e_i)_i \in T_I$ cannot decrease too much. Specifically, denoting the current set of paths by $P'_0 = P'$, the new average $m(\cdot)$ values satisfy

$$m'_I := \mathbb{E}_{(e_i)\in T_I}[m_{P'_0}((e_i)_i)] = |P'_0|/|T_I| \geq \frac{|P'_I|/|T_I|}{((k'-1)\log\Delta)^{2^{k'}-2}} \geq \frac{m_I}{2((k'-1)\log\Delta)^{2^{k'}-2}}.$$

Finally, while there exists any tuple $(e_i) \in \prod_{i\in I} E_i$ for any $I \subseteq [k']$ such that $m_{P'}((e_i)) \leq 2^{-k} m'_I$, remove all paths corresponding to this tuple from P'. The total number of paths which may be pruned at this stage is at most

$$\sum_{\emptyset\neq I\subseteq[k]} |T_I| \cdot 2^{-k} m'_I = \sum_{\emptyset\neq I\subseteq[k]} 2^{-k}|P'_0| = (2^k-1)2^{-k}|P'_0|.$$

Thus, the final set P' satisfies

$$|P'| \geq 2^{-k}|P'_0| \geq \frac{|P|}{2^k((k'-1)\log\Delta)^{2^{k'}-1}}.$$

Furthermore, for every I-tuple $(e_i)_{i\in I}$ which is involved in any path in P', we have $m((e_i)) \leq m_I$ and

$$m((e_i)) \geq 2^{-k} m'_I \geq \frac{m_I}{2^{k+1}((k'-1)\log\Delta)^{2^{k'}-2}}. \qquad \square$$

3 Hardness of approximation

Our reductions are based on the framework developed by [22, 20]. 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.[5]

[5]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*.

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 $\varepsilon > 0$, we say that an instance of MIN-REP is a YES instance if $\mathrm{OPT} = 2r$ (i. e., a single node is chosen from each group) and is a NO instance if $\mathrm{OPT} \geq 2^{\log^{1-\varepsilon} n} r$. We will sometimes refer to the hardness gap (in this case $2^{\log^{1-\varepsilon} n}$) as the *soundness* s, due to the connection between MIN-REP and proof systems. The following theorem is due to Kortsarz [22] (the polynomial relations between the parameters are implicit rather than explicit in his proof, but are straightforward to verify since the instances used in [22] are obtained by parallel repetition [27] applied to instances of 3SAT-5 which have a constant gap [3, 2]).

Theorem 3.1 ([22]). *Unless* $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{\mathrm{polylog}(n)})$, *for any constant* $\varepsilon > 0$ *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 [20] and [22] 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 first consider the directed setting (note that here the "degree" we are trying to minimize is the sum of the in-degree and the out-degree). Suppose we are given a bipartite MIN-REP instance $\widetilde{G} = (A, B, \widetilde{E})$ with associated supergraph $G' = (U, V, E')$ from Theorem 3.1. 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 $k_L = \lfloor (k-1)/2 \rfloor$, and let $k_R = \lceil (k-1)/2 \rceil$. The four sets of vertices are

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

The actual vertex set V_G of our LDkS instance G will be $V_{\text{out}}^L \cup V_{\text{out}}^R \cup V_{\text{in}}^L \cup V_{\text{in}}^R$. 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.

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

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.)

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

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

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

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

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

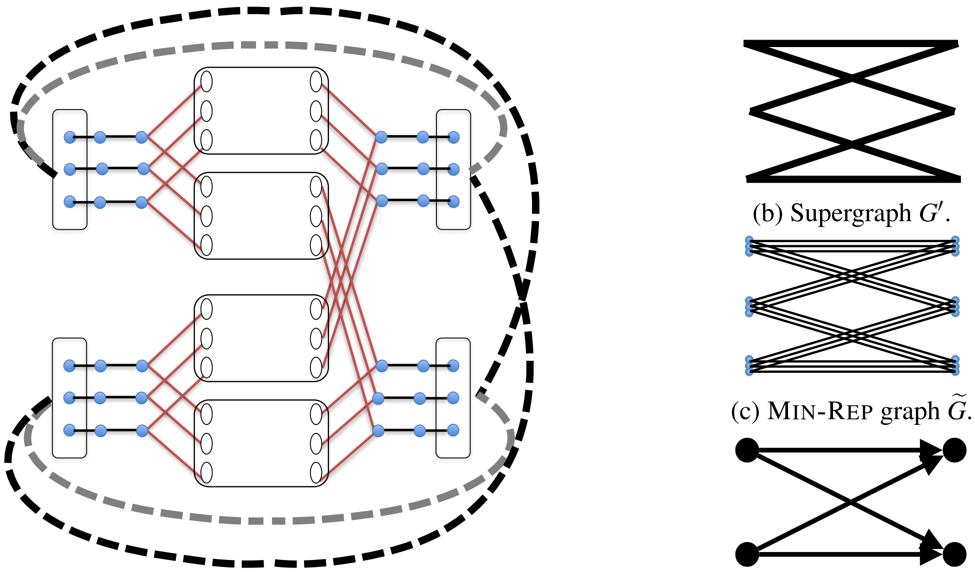
and call the edges in E_{path}^L and E_{path}^R *path-edges*. Finally, for technical reasons we need to add *group edges* internally in each group in each copy of $\widetilde{G}$: let

$$E_{\text{group}}^L = \{((a,e),(a',e)) : e \in E_H \wedge a,a' \in \Gamma(u) \text{ for some } u \in U\}\,, \quad \text{and}$$
$$E_{\text{group}}^R = \{((b,e),(b',e)) : e \in E_H \wedge b,b' \in \Gamma(v) \text{ for some } v \in V\}\,.$$

Our final edge set is the union of all of these, namely

$$E_{\text{out}} \cup E_{\text{in}} \cup E_{\text{con}}^L \cup E_{\text{con}}^R \cup E_{\text{path}}^L \cup E_{\text{path}}^R \cup E_{\text{group}}^L \cup E_{\text{group}}^R\,.$$

A small example of this construction appears as Figure 1.



(a) Final graph G.

(b) Supergraph G'.

(c) MIN-REP graph $\widetilde{G}$.

(d) $H = K_{2,2}$.

Figure 1: Graph G created by the reduction for $k = 7$ when using supergraph G' from Figure 1b and using $H = K_{2,2}$ (Figure 1d). Blue vertices are outer nodes, and the inner nodes are contained in the clear central ovals (individual nodes not pictured), each of which is a group from the MIN-REP graph $\widetilde{G}$ (Figure 1c). Each dotted edge represents a copy of G' (so six actual edges), the union of which are the outer edges. Each of the four central rectangles represents a copy of the MIN-REP graph $\widetilde{G}$, and together these are the inner edges. Each small oval is a clique (the group edges), and the edges forming the outer 3-paths are the path-edges. Each red edge from the outer nodes to the inner nodes actually represents a star, from the outer node to the associated inner group. These stars together form the connection edges.

3.1.1 Analysis

We begin by simply figuring out the degrees of each kind of node in our construction. Each inner node is incident on $2|\Sigma|$ group edges, a single connection edge, and at most $d_{\widetilde{G}} \leq d_{G'}|\Sigma|$ inner edges, for a total degree of at most $d_{G'}|\Sigma| + 2|\Sigma| + 1$. Minimal outer nodes are each incident on 1 path-edge and $|\Sigma|d_H$ connection edges, for a total degree of $|\Sigma|d_H + 1$. Maximal outer nodes are incident on 1 path-edge and $d_{G'}d_H$ outer edges, for a total degree of $d_{G'}d_H + 1$. If $k = 3$ of 4 then there are outer nodes which are both maximal and minimal and thus have degree $|\Sigma|d_H + d_{G'}d_H$, but this will not matter. Together with our choice of d_H, this gives the following lemma.

Lemma 3.2. *The maximum degree in G is achieved at either the maximal outer or minimal outer nodes, and is at most $O(d_{G'}^2 + |\Sigma|^2)$.*

We now 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* if it 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 to span outer edges (other than using those edges to span themselves). 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 3.3. *If $\widetilde{G}$ is 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 $\widetilde{G}$ 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 $\{f(u), f(v)\} \in \widetilde{E}$ 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 $((u,x,1),(a,(x,y))) \in E^L_{\text{con}}$. 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 $(a,e) - (f(u),e) - (f(v),e) - (b,e)$, 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 $\{f(u), f(v)\} \in \widetilde{E}$ 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 3.4. *If $\widetilde{G}$ is a NO instance of* 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 $\widetilde{G}$ 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 $E^L_{\text{con}} \cap E(\hat{G})$ we add it to $B_{(x,y)}$, as well as each connection edge $((b,(x,y)),(v,y,1)) \in E^R_{\text{con}} \cap E(\hat{G})$. 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 $((u,x,1),(a,(x,y))) \in E^L_{\text{con}} \cap B_{(x,y)}$ we will include a and for each edge $((b,(x,y)),(v,y,1)) \in E^R_{\text{con}} \cap B_{(x,y)}$ 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 set 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 at most $s(|U|+|V|)$. □

We can now use Lemma 3.3 and Lemma 3.4 to prove the desired hardness for Directed LDkS.

Theorem 3.5. *Unless $\mathsf{NP} \subseteq \mathsf{DTIME}(2^{\text{polylog}(n)})$, 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. Lemma 3.3 and Lemma 3.4, when combined with Theorem 3.1, imply hardness of $\Omega(s)$. We know from Lemma 3.2 that $\Delta = O(d^2_{G'} + |\Sigma|^2)$. Since we specifically chose to use hard MIN-REP instances where $d_{G'}$ and $|\Sigma|$ are polynomial in s (see Theorem 3.1), this proves the theorem. □

3.2 Undirected LDkS

We now want to handle the undirected case. 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 are extra options. Most of them can easily be ruled out by the fact that a spanning path must have length at most k. For example, this implies that an outer edge cannot be spanned by a path which uses more then one inner edge (in the directed setting this was clear from the directions). But there is one particularly problematic 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^R_{out}. 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 [14], giving the following lemma. As it is essentially straightforward from [14], we give only the outline of the proof. First, we will need the following definitions. The MAX-REP problem is the maximization version of MIN-REP: instead of finding the smallest possible REP-cover, we allow ourselves to pick at most one node from each group in the MIN-REP graph and attempt to maximize the number of covered superedges. The MIN-REP$_k$ problem is MIN-REP restricted to instances where the supergirth is larger than k. The following theorem appears as Theorem 5 in [14] (the extra condition on superdegrees is Lemma 3 from [14]).

Theorem 3.6 ([14]). *Suppose that there is no (randomized) polynomial-time algorithm that, given an instance of* MAX-REP *in which $|U| = |V| = n/2$ and all supervertices have superdegree d, can distinguish between the following cases.*

1. *All superedges are satisfiable, and*

2. *At most an $s \leq 1/(16 \log n)$ fraction of the superedges are satisfiable.*

Then there is some constant c so that for all α with $16 \log n \leq \alpha \leq \min\{1/s, d^{1/k}/\log|\Sigma|\}$, there is no (randomized) polynomial-time algorithm that, given an instances of MIN-REP$_k$ *where all supernodes have superdegree within a factor of two of $\alpha \log|\Sigma|$, can distinguish between the following cases.*

1. *The smallest REP-cover has size at most n, and*

2. *The smallest REP-cover has size at least $n\sqrt{\alpha}/c$,*

where n is the size of the supergraph.

The following lemma gives the MIN-REP instances which we will use through the rest of the section.

Lemma 3.7. *Unless* $\mathsf{NP} \subseteq \mathsf{BPTIME}(n^{\mathrm{polylog}(n)})$, *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. *$d_{G'} \geq |\Sigma|$.*

Proof. We will start out with the MAX-REP instances derived from applying parallel repetition [27] to 3SAT-5, but where the graph is balanced and regular by including 3 copies of the variable nodes and 5 copies of the clause nodes (see Section 3 of [14] for a slightly more formal description). These instances have degree $d = 15^{\ell}$ and $|\Sigma| = 7^{\ell}$ and soundness $s = 2^{\ell/c}$ for some value ℓ (the number of times parallel repetition is applied) and constant c (the loss in parallel repetition). (Note that in the context of MAX-REP the soundness is less than 1, since it represents the fraction of superedges that can be covered in a NO instance.) Now if we try to use [14] to get the supergirth larger than $k+1$ we get into trouble, since the degrees (and the soundness s) are reduced to $15^{\ell/k}$, while the alphabet size $|\Sigma| = 7^{\ell}$ is much larger.

We get around this by boosting the degree, in much the same way as in the directed case. We create a new MAX-REP instance by taking the cross product of the original instance and the complete bipartite graph with d^k nodes on each side. In other words, we create a new instance in which there are d^k copies of each side of the original MAX-REP instance, and each left copy and right copy (with the edges between them) form a copy of the original instance. Clearly every node now has degree $d' = 15^{\ell} \cdot 15^{\ell k} = 15^{\ell(k+1)} = d^{k+1}$, while $|\Sigma|$ is unchanged. It is not hard to see that the soundness s is also unchanged, so is still $2^{\ell/c}$. (This argument appears in detail in [14].)

We can now apply Theorem 3.6 with $\alpha = (d')^{1/(k+1)}/\log|\Sigma|$ to these instances, which gives the claimed bounds. □

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 $n_H d_H \leq n_H^{1+(1/(3k^2))}$ [5]). 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 3.7). We will then subsample in essentially the same way as [14]: for every outer edge $\{(u,x,k_L),(v,y,k_R)\}$ we will flip an independent coin, keeping the edge with probability $p = \alpha/d$ and removing it with probability $1-p$ (we will set $\alpha = d^{(k+2)/(2(k+1))}/4$, so $p = 1/(4d^{k/(2(k+1))})$).[6] 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 relatively few bad edges in G_α.

[6] While the combination of defining both p and α may seem redundant, we include both in order to be consistent with [14], as well as for the convenience of having α be closely related to the average degree.

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 3.3.

Lemma 3.8. *If $\widetilde{G}$ is a YES instance of* MIN-REP *from Lemma 3.7, then there is a k-spanner of $\widehat{G_\alpha}$ with maximum degree at most $3d_H + 1$.*

Proof. We use the same spanner construction as in Lemma 3.3, which proved that the maximum degree is at most $\max\{d_{G'} + 2|\Sigma| + 1, d_H + 1\}$. The only difference is that now we need to replace the use of $d_{G'}$ in Lemma 3.3 with the maximum superdegree of any of the $|E_H|$ instances of MIN-REP that we are left with after our sampling to get $\widehat{G}_\alpha$. However, since this sampling only decreases degrees, the maximum degree is still at most $\max\{d_{G'} + 2|\Sigma| + 1, d_H + 1\} \le \max\{3d_{G'} + 1, d_H + 1\} = 3d_H + 1$. □

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 sampled instances of MIN-REP (i. e., why we had to start with instances based on Lemma 3.7 rather than generic hard MIN-REP instances, like those from Theorem 3.1).

Throughout the rest of this section we will analyze paths of length exactly k, in order to simplify notation. The analysis of paths of length less than k is identical, just with k replaced by some $k' < k$.

Lemma 3.9. *For any outer edge, the number of bad paths is at most $O(4^k d^{(k-1)/2})$.*

Proof. Let $\{(u,x),(v,y)\}$ be an outer edge. For now we will only analyze paths of length exactly k. Then any bad path has the form

$$(u,x) = (u_0,x_0) - (v_0,y_0) - (u_1,x_1) - \cdots - (u_{(k-1)/2}, x_{(k-1)/2}) - (v_{(k-1)/2}, y_{(k-1)/2}) = (v,y)\,.$$

Consider the projection of this path on the first coordinate, i. e., the path

$$u_0 - v_0 - u_1 - v_1 - \cdots - u_{(k-1)/2} - v_{(k-1)/2}\,.$$

Clearly this is a (not necessarily simple) path in G'. Call this path p. In fact it cannot be a simple path, since together with the edge $\{u,v\}$ it would form a cycle of length $k+1$ in G', which cannot happen since we know that G' has girth at least $k+2$. Moreover, p must include $\{u,v\}$ at some point, or else by shortcutting we would be able to get a cycle of length at most $k+1$. But we have even more: every edge in this path other than $\{u,v\}$ must appear an even number of times (in an undirected sense, so it might be traversed once in one direction and a second time in the other direction). To see this, suppose that at some point this path went from u_i to v_i but then never went across this edge in the opposite direction. Then by shortcutting to get simple paths, we know that there is a path from u_0 to u_i and a path from v_i to $v_{(k-1)/2}$ with combined length less than k. Together with the existence of the $\{u,v\}$ edge, this clearly implies the existence of a cycle of length at most $k+1$. This contradicts our starting assumption that G' has girth larger than $k+1$.

Since every edge other than $\{u,v\}$ must be used in both directions, we know that there are at most $(k-1)/2$ hops of p where an edge is used for an odd-numbered time (e. g., for the first time, or third time, etc). In order to bound the number of possible paths p, consider a hop $\gamma \in [k]$ of p in which an edge is traversed an odd-numbered time, and suppose that p is currently at some node x (e. g., if $\gamma = 5$ then after four hops the path is at x and the fifth hop involves traversing an edge an odd-numbered time). There are clearly at most $d_{G'}$ choices for *which* edge appears at hop α, since that is the degree of x. On the other hand, suppose that at hop α we traverse an edge for an even-numbered time. Then it is easy to see by induction that we only have one possible choice, i. e., there is only one edge incident on x which has already been crossed an odd number of times. (Clearly this is true initially, and if we ever traverse an edge to a node y so that y has at least two incident edges that have been traversed an even number of times then we would have found a cycle of length at most k.)

Thus each path p is uniquely identified by two pieces of information: which hops traverse an edge for an odd-numbered time, and which of the $d_{G'}$ possible edges are traversed in those hops. There are at most

$$\binom{k}{(k-1)/2} \leq 2^k$$

possibilities for the first piece of information, and at most $d_{G'}^{(k-1)/2}$ possibilities for the second. Thus there are a total of at most $2^k d_{G'}^{(k-1)/2}$ possible paths p.

Since H also has girth larger than $k+1$, the same analysis holds for our bad path projected on the second coordinate but with $d_{G'}$ replaced by d_H. Thus the total number of bad paths is at most $2^k d_{G'}^{(k-1)/2} \cdot 2^k d_H^{(k-1)/2} = 4^k d^{(k-1)/2}$. □

Lemma 3.9 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 3.10. *Let $e = \{(u,x,k_L),(v,y,k_R)\}$ be an outer edge in G. The probability that it is bad in G_α and exists in G_α is at most $4^k \alpha^{k+1}/d^{(k+3)/2}$.*

Proof. Fix some bad path in G between the endpoints of e. The probability that it survived the sampling is at most $p^k = (\alpha/d)^k$. We can now use Lemma 3.9 and a union bound to get that the total probability that e is bad is at most $4^k d^{(k-1)/2} \cdot (\alpha/d)^k = (4\alpha)^k/(d^{(k+1)/2})$. The probability that e itself survived the sampling is α/d, and this is independent of whether or not any of the bad paths survive. Thus the total probability that e is bad and is in G_α is at most $4^k \alpha^{k+1}/d^{(k+3)/2}$ as claimed. □

Lemma 3.11. *With probability at least $3/4$ the number of outer edges in G_α that are bad is at most $|U| \cdot |X| \cdot d_H$.*

Proof. There are at most $|U| \cdot |X| \cdot d$ outer edges in G, and from Lemma 3.10 we know that each is bad with probability at most $4^k \alpha^{k+1}/d^{(k+3)/2}$ (not necessarily independently). Thus the expected number of bad edges in G_α is at most

$$|U| \cdot |X| \cdot d \cdot \frac{4^k \alpha^{k+1}}{d^{(k+3)/2}} = \frac{|U| \cdot |X| \cdot 4^k \alpha^{k+1}}{d^{(k+1)/2}}.$$

Now applying Markov's inequality implies that with probability at least $3/4$ the number of bad edges in G_α is at most

$$\frac{|U|\cdot|X|\cdot 4^{k+1}\alpha^{k+1}}{d^{(k+1)/2}}.$$

When we plug in our choice of $\alpha = d^{(k+2)/(2(k+1))}/4$, this gives us at most $|U|\cdot|X|\cdot d^{1/2} = |U|\cdot|X|\cdot d_H$ bad edges in G_α, as claimed. □

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. Let $A \subseteq E_H$ be the edges in H which correspond to these instances. It is well-known that removing only $O(|U|)$ 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., the proof of [14, Theorem 5]—intuitively, each superedge can only decrease the size of the optimal REP-cover by at most 2). Therefore, $\widehat{G_\alpha}$ has the same soundness as G_α, up to a constant factor. Now that there are no bad edges, the only way to span an outer edge in $\widehat{G_\alpha}$ is either the edge itself or a canonical path. This means that we can now analyze this case just like the directed case (with the main difference being that we can only use $|A| \geq |E_H|/2$ of the $|E_H|$ MIN-REP instances to prove our bound, though this is sufficient). This implies that in a NO instance all spanners must have large maximum degree, through an analysis similar to Lemma 3.4.

Lemma 3.12. *If $\widetilde{G}$ is a NO instance of* MIN-REP*, then every k-spanner of G has maximum degree at least $\widetilde{\Omega}(d_H \cdot d_{G'}^{1/(2(k+1))})$.*

Proof. Since $\widehat{G_\alpha}$ has no bad edges, every outer edge must be spanned by either itself or by a canonical path. This is the same as in the directed case, and the same analysis from Lemma 3.4 continues to hold. Let d'_H denote the average degree of a node in H restricted to A. This is clearly at least $d_H/2$. Now using the analysis of Lemma 3.4, we know that if all of the MIN-REP instances of A require REP-covers of size at least $s'(|U|+|V|)$ then the maximum degree must be at least $\Omega(s'd'_H) = \Omega(s'd_H)$.

So it remains simply to bound s'. But this is straightforward, thanks to [14], which proved that if we start with a MIN-REP instance with soundness s, and then independently subsample to make the average degree $\alpha' \geq \Omega(\log n)$ (with $\alpha' \leq 1/s$) and then remove a linear number of superedges, we are left with a MIN-REP instance where the soundness is $s' = \widetilde{\Omega}(\sqrt{\alpha'})$ (see [14, Lemma 5] and the proof of Theorem 3.6 in particular). In our case,

$$\alpha' = pd_{G'} = \frac{\alpha}{d} d_{G'} = \frac{d^{(k+2)/(2(k+1))}}{4d} d_{G'} = \frac{1}{4d^{k/(2(k+1))}} d_{G'} = \frac{d_{G'}}{4d_{G'}^{k/(k+1)}} = \Omega(d_{G'}^{1/(k+1)}).$$

Hence s' is at least $\widetilde{\Omega}\left(d_{G'}^{1/(2(k+1))}\right)$. This proves the lemma. □

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

Theorem 3.13. *Unless* $\mathsf{NP} \subseteq \mathsf{BPTIME}(n^{\mathrm{polylog}(n)})$*, 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 $k \geq 3$).*

Proof. We know from Lemma 3.8 and Lemma 3.12 that we have a gap of $d_{G'}^{1/(2(k+1))}$. So we simply need to argue that $d_{G'}^{1/2(k+1)}$ is at least $\Delta^{\Omega(1/k)}$. The maximum degree in $\widehat{G_\alpha}$ is at most the maximum degree in G_α which is at most the maximum degree in G. By Lemma 3.2 and the fact that $d_{G'} \geq |\Sigma|$ by Lemma 3.7, this is at most $O(d_{G'}^2)$. This proves the theorem. □

4 Open problems

The obvious open problem is to close the gap between the upper bound of $\widetilde{O}(\Delta^{(1-1/k)^2})$ and the lower bound of $\Delta^{\Omega(1/k)}$ for LDkS. It is unclear what the true behavior of the approximability of LDkS is as a function of k, and even whether the problem gets easier or harder as k gets larger (or if the approximability stays the same). Our current upper bound gets larger with k, while our current lower bound gets smaller. What is the true behavior?

Another interesting set of questions involves relaxing the notion of approximation to allow bicriteria approximations, where we return a spanner with stretch slightly larger than k but compare ourselves to the lowest-degree k-spanner. Our current hardness results break down when we allow a constant violation of the stretch, so (for example) we cannot rule out the possibility of an algorithm which returns a 9-spanner with maximum degree at most a constant times the maximum degree of the best 3-spanner. Obtaining good bicriteria approximations, or proving that they cannot exist, is an extremely interesting area for future research (for both LDkS and other spanner approximation problems).

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. [doi:10.1007/BF02189308] 2, 4

[2] 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, 1998. Preliminary version in FOCS'92. [doi:10.1145/278298.278306] 16

[3] SANJEEV ARORA AND SHMUEL SAFRA: Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. Preliminary version in FOCS'92. [doi:10.1145/273865.273901] 16

[4] SURENDER BASWANA, TELIKEPALLI KAVITHA, KURT MEHLHORN, AND SETH PETTIE: Additive spanners and (α,β)-spanners. *ACM Trans. Algorithms*, 7(1):5:1–5:26, 2010. Preliminary version in SODA'05. [doi:10.1145/1868237.1868242] 4

[5] MOHSEN BAYATI, ANDREA MONTANARI, AND AMIN SABERI: Generating random graphs with large girth. In *Proc. 20th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'09)*, pp. 566–575. ACM Press, 2009. ACM DL. [arXiv:0811.2853] 22

[6] PIOTR BERMAN, ARNAB BHATTACHARYYA, KONSTANTIN MAKARYCHEV, SOFYA RASKHODNIKOVA, AND GRIGORY YAROSLAVTSEV: Improved approximation for the directed spanner problem. In *Proc. 38th Internat. Colloq. on Automata, Languages and Programming (ICALP'11)*, volume 6755 of *LNCS*, pp. 1–12. Springer, 2011. [doi:10.1007/978-3-642-22006-7_1, arXiv:1012.4062] 2, 3, 5, 6

[7] ARNAB BHATTACHARYYA, ELENA GRIGORESCU, KYOMIN JUNG, SOFYA RASKHODNIKOVA, AND DAVID P. WOODRUFF: Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012. Preliminary version in SODA'09. [doi:10.1137/110826655, arXiv:0808.1787] 5

[8] T.-H. HUBERT CHAN, MICHAEL DINITZ, AND ANUPAM GUPTA: Spanners with slack. In *Proc. 14th Ann. European Symp. on Algorithms (ESA'06)*, pp. 196–207. Springer, 2006. [doi:10.1007/11841036_20] 4

[9] BARUN CHANDRA, GAUTAM DAS, GIRI NARASIMHAN, AND JOSÉ SOARES: New sparseness results on graph spanners. *Internat. J. Comput. Geom. Appl.*, 5(1):125–144, 1995. Preliminary version in SoCG'92. [doi:10.1142/S0218195995000088] 2

[10] SHIRI CHECHIK: New additive spanners. In *Proc. 24th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'13)*, pp. 498–512. ACM Press, 2013. [doi:10.1137/1.9781611973105.36] 4

[11] SHIRI CHECHIK, MICHAEL LANGBERG, DAVID PELEG, AND LIAM RODITTY: Fault tolerant spanners for general graphs. *SIAM J. Comput.*, 39(7):3403–3423, 2010. Preliminary version in STOC'09. [doi:10.1137/090758039] 4

[12] EDEN CHLAMTÁČ AND MICHAEL DINITZ: Lowest degree k-spanner: Approximation and hardness. In *Proc. 17th Internat. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'14)*, volume 28 of *LIPIcs*, pp. 80–95. Springer, 2014. [doi:10.4230/LIPIcs.APPROX-RANDOM.2014.80] 1

[13] EDEN CHLAMTÁČ, MICHAEL DINITZ, AND ROBERT KRAUTHGAMER: Everywhere-sparse spanners via dense subgraphs. In *Proc. 53rd FOCS*, pp. 758–767. IEEE Comp. Soc. Press, 2012. [doi:10.1109/FOCS.2012.61, arXiv:1205.0144] 2, 3, 5

[14] 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, 2016. Preliminary version in ICALP'12. [doi:10.1145/2818375, arXiv:1203.0224] 4, 5, 21, 22, 25

[15] MICHAEL DINITZ AND ROBERT KRAUTHGAMER: Directed spanners via flow-based linear programs. In *Proc. 43rd STOC*, pp. 323–332. ACM Press, 2011. [doi:10.1145/1993636.1993680, arXiv:1011.3701] 3, 5

[16] MICHAEL DINITZ AND ROBERT KRAUTHGAMER: Fault-tolerant spanners: Better and simpler. In *Proc. 30th Symp. on Principles of Distributed Comput. (PODC'11)*, pp. 169–178. ACM Press, 2011. [doi:10.1145/1993806.1993830, arXiv:1101.5753] 4, 5

[17] YEVGENIY DODIS AND SANJEEV KHANNA: Designing networks with bounded pairwise distance. In *Proc. 31st STOC*, pp. 750–759. ACM Press, 1999. [doi:10.1145/301250.301447] 5

[18] MICHAEL ELKIN AND DAVID PELEG: Strong inapproximability of the basic k-spanner problem. In *Proc. 27th Internat. Colloq. on Automata, Languages and Programming (ICALP'00)*, pp. 636–647. Springer, 2000. [doi:10.1007/3-540-45022-X_54] 4

[19] MICHAEL ELKIN AND DAVID PELEG: Approximating k-spanner problems for $k > 2$. *Theoret. Comput. Sci.*, 337(1-3):249–277, 2005. Preliminary version in IPCO'01. [doi:10.1016/j.tcs.2004.11.022] 5

[20] MICHAEL ELKIN AND DAVID PELEG: The hardness of approximating spanner problems. *Theory Comput. Sys.*, 41(4):691–729, 2007. Preliminary version in STACS'00. [doi:10.1007/s00224-006-1266-2] 4, 5, 15, 16

[21] MICHAEL ELKIN AND SHAY SOLOMON: Fast constructions of lightweight spanners for general graphs. *ACM Trans. Algorithms*, 12(3):29:1–29:21, 2016. Preliminary version in SODA'13. [doi:10.1145/2836167, arXiv:1207.1668] 4

[22] GUY KORTSARZ: On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001. Preliminary version in APPROX'98. [doi:10.1007/s00453-001-0021-y] 4, 5, 15, 16

[23] GUY KORTSARZ AND DAVID PELEG: Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994. Preliminary version in SWAT'92. [doi:10.1006/jagm.1994.1032] 4

[24] GUY KORTSARZ AND DAVID PELEG: Generating low-degree 2-spanners. *SIAM J. Comput.*, 27(5):1438–1456, 1998. Preliminary version in SODA'94. [doi:10.1137/S0097539794268753] 2, 3, 4, 5, 6

[25] DAVID PELEG AND ALEJANDRO A. SCHÄFFER: Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989. [doi:10.1002/jgt.3190130114] 2, 4

[26] DAVID PELEG AND JEFFREY D. ULLMAN: An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989. Preliminary version in PODC'87. [doi:10.1137/0218050] 2, 4

[27] RAN RAZ: A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. Preliminary version in STOC'95. [doi:10.1137/S0097539795280895] 16, 22

[28] MOHIT SINGH AND LAP CHI LAU: Approximating minimum bounded degree spanning trees to within one of optimal. *J. ACM*, 62(1):1:1–1:19, 2015. Preliminary version in STOC'07. [doi:10.1145/2629366] 2

[29] MIKKEL THORUP AND URI ZWICK: Compact routing schemes. In *Proc. 13th Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA'01)*, pp. 1–10. ACM Press, 2001. [doi:10.1145/378580.378581] 2

[30] MIKKEL THORUP AND URI ZWICK: Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005. Preliminary version in STOC'01. [doi:10.1145/1044731.1044732] 4

AUTHORS

Eden Chlamtáč
Assistant Professor
Ben Gurion University, Beersheva, Israel
chlamtac@cs.bgu.ac.il
https://www.cs.bgu.ac.il/~chlamtac/

Michael Dinitz
Assistant Professor
Johns Hopkins University, Baltimore, MD
mdinitz@cs.jhu.edu
http://www.cs.jhu.edu/~mdinitz/

ABOUT THE AUTHORS

EDEN CHLAMTÁČ graduated from Princeton University in Princeton, NJ, in 2008; his advisor was Sanjeev Arora. He then spent two years as a postdoc at the Weizmann Institute of Science, and one and three semesters as a postdoc at the University of Toronto and Tel-Aviv University, respectively, before joining the faculty at Ben Gurion University. If memory serves, when he used to have spare time he would enjoy playing piano and studying languages which rarely did him much good.

MICHAEL DINITZ graduated from the Computer Science Department at Carnegie Mellon University in 2010, where he was advised by Anupam Gupta. He then spent three years as a postdoc at the Weizmann Institute of Science, hosted by Robert Krauthgamer and David Peleg, before joining the faculty at Johns Hopkins University. He is interested in approximation algorithms and applications to computer networking, and particularly in network design problems and problems involving metric spaces.