Improved Approximation for the Directed Spanner Problem

Piotr Berman[1], Arnab Bhattacharyya[2⋆], Konstantin Makarychev[3], Sofya Raskhodnikova[1⋆⋆], and Grigory Yaroslavtsev[1⋆⋆⋆]

[1] Pennsylvania State University. {berman, sofya, grigory}@cse.psu.edu.
[2] Massachusetts Institute of Technology. abhatt@mit.edu.
[3] IBM T.J. Watson Research Center. konstantin@us.ibm.com.

Abstract. We give an $O(\sqrt{n}\log n)$-approximation algorithm for the problem of finding the sparsest spanner of a given *directed* graph G on n vertices. A spanner of a graph is a sparse subgraph that approximately preserves distances in the original graph. More precisely, given a graph $G = (V, E)$ with nonnegative edge lengths $d : E \to \mathbb{R}^{\geq 0}$ and a *stretch* $k \geq 1$, a subgraph $H = (V, E_H)$ is a *k-spanner* of G if for every edge $(u, v) \in E$, the graph H contains a path from u to v of length at most $k \cdot d(u, v)$. The previous best approximation ratio was $\tilde{O}(n^{2/3})$, due to Dinitz and Krauthgamer (STOC '11).
We also present an improved algorithm for the important special case of directed 3-spanners with unit edge lengths. The approximation ratio of our algorithm is $\tilde{O}(n^{1/3})$ which almost matches the lower bound shown by Dinitz and Krauthgamer for the integrality gap of a natural linear programming relaxation. The best previously known algorithms for this problem, due to Berman, Raskhodnikova and Ruan (FSTTCS '10) and Dinitz and Krauthgamer, had approximation ratio $\tilde{O}(\sqrt{n})$.

1 Introduction

A spanner of a graph is a sparse subgraph that approximately preserves pairwise distances in the original graph. This notion was first used by Awerbuch [2] and explicitly introduced by Peleg and Schäffer [23].

Definition 1.1 (k-spanner, [2, 23]). *Given a graph $G = (V, E)$ with nonnegative edge lengths $d : E \to \mathbb{R}^{\geq 0}$ and a real number $k \geq 1$, a subgraph $H = (V, E_H)$ is a **k-spanner** of G if for all edges $(u, v) \in E$, the graph H contains a path from u to v of length at most $k \cdot d(u, v)$. The parameter k is called the **stretch**.*

Spanners have numerous applications, such as efficient routing [9, 10, 25, 27, 28], simulating synchronized protocols in unsynchronized networks [24], parallel, distributed and streaming algorithms for approximating shortest paths [7, 8, 13, 18],

⋆ Supported by National Science Foundation grants CCF-1065125 and CCF-0728645.
⋆⋆ Supported by National Science Foundation (NSF/CCF CAREER award 0845701).
⋆⋆⋆ Supported by NSF / CCF CAREER award 0845701, University Graduate Fellowship and College of Engineering Fellowship.

algorithms for distance oracles [3, 29], property testing, property reconstruction and key management in access control hierarchies (see [6, 5, 20], the survey in [26] and references therein).

We study the computational problem of finding the sparsest spanner of a given *directed* graph G and stretch k, that is, a k-spanner of G with the smallest number of edges. We refer to this problem as DIRECTED k-SPANNER and distinguish between the case of unit edge lengths (i.e., $d(e) = 1$ for all $e \in E$) and arbitrary edge lengths. The UNDIRECTED k-SPANNER problem refers to the task of finding the sparsest k-spanner of a given undirected graph. The natural reduction from UNDIRECTED k-SPANNER to DIRECTED k-SPANNER preserves approximation ratio.

Our main results are an algorithm with approximation ratio $O(\sqrt{n} \log n)$ for DIRECTED k-SPANNER with arbitrary edge lengths and an algorithm with approximation ratio $O(n^{1/3} \log^2 n)$ for DIRECTED 3-SPANNER with unit edge lengths, where n is the number of nodes in the input graph G. Our approximation guarantee for DIRECTED 3-SPANNER almost matches the integrality gap of $\Omega(n^{1/3-\epsilon})$ of Dinitz and Krauthgamer [11] for a natural linear programming relaxation of the problem. Our result also directly implies the same approximation ratio for the UNDIRECTED 3-SPANNER problem with unit edge lengths.

Relation to Previous Work. DIRECTED k-SPANNER with *unit edge lengths* has been extensively studied. Note that in this case, we can assume that k is a positive integer. For $k = 2$, the problem has been completely resolved: Kortsarz and Peleg [21] and Elkin and Peleg [15] gave $O(\log n)$-approximation, and Kortsarz [22] proved that the approximation cannot be improved unless P=NP. Elkin and Peleg [14] gave $\tilde{O}(n^{2/3})$-approximation for DIRECTED 3-SPANNER. For general $k \geq 3$, Bhattacharyya *et al.* [6] presented $\tilde{O}(n^{1-1/k})$-approximation; then, Berman, Raskhodnikova and Ruan [4] improved it to $\tilde{O}(n^{1-1/\lceil k/2 \rceil})$, and recently Dinitz and Krauthgamer [11] gave $\tilde{O}(n^{2/3})$-approximation, presenting the first algorithm with approximation ratio independent of k. For the special cases of $k = 3$ and $k = 4$, Berman, Raskhodnikova and Ruan showed an $\tilde{O}(\sqrt{n})$-approximation. Independently, Dinitz and Krauthgamer also gave an $\tilde{O}(\sqrt{n})$-approximation for the case $k = 3$. Thus, our algorithms improve on [4] for all $k \geq 3$, where $k \neq 4$, and on [11] for all $k \geq 3$.

Dinitz and Krauthgamer gave the first approximation algorithm for the problem for arbitrary edge lengths. For this case, one can no longer assume that k is an integer. Dinitz and Krauthgamer achieved $\tilde{O}(n^{2/3})$-approximation for arbitrary edge lengths for all $k > 1$. We improve this approximation to $\tilde{O}(n^{1/2})$ for all $k > 1$.

Spanners for undirected graphs behave somewhat differently in terms of their approximability. For all integer k and for all undirected graphs G with arbitrary edge lengths, it is known [23, 1] that a k-spanner of G with at most $n \cdot \lceil n^{2/(k+1)} \rceil$ edges can be constructed in polynomial time. Since a k-spanner of a connected graph must have at least $n - 1$ edges, an approximation ratio of $O(n^{2/(k+1)})$ trivially follows. In particular, for $k = 3$, this argument yields an approximation ratio of $O(\sqrt{n})$. Our result improves the ratio for UNDIRECTED 3-SPANNER to $\tilde{O}(n^{1/3})$ in the case of unit-length edges.

Elkin and Peleg [14, 17], improving on [22], showed that it is quasi-NP-hard to approximate DIRECTED k-SPANNER, even when restricted to unit edge lengths, with ratio better than $2^{\log^{1-\epsilon} n}$ for $k \in (3, n^{1-\delta})$ and all $\delta, \epsilon \in (0, 1)$. For UNDIRECTED k-SPANNER with unit-length edges, such a strong hardness result does not hold since the problem is $O(1)$-approximable when $k = \Omega(\log n)$. However, for constant $k \geq 3$, it is still quasi-NP-hard to approximate with a ratio better than $2^{\log^{1-\epsilon} n}$ [14, 12]. When the edge lengths are arbitrary, the same inapproximability also holds for $k \in (1, 3)$, even for the undirected case [17].

Our Techniques. Our algorithms operate by combining two graphs: the first obtained from randomized rounding of a fractional solution to a flow-based linear programming relaxation of the problem and the second obtained by growing shortest-path trees from randomly selected vertices. The idea of combining a linear programming approach with sampling to solve DIRECTED k-SPANNER first appeared in [6]. Dinitz and Krauthgamer [11] used the same approach, but with a novel, flow-based linear program (LP). Our main insight is to use randomized LP rounding schemes. We also give a new LP relaxation, slightly simpler than that in [11]. In the case of unit edge lengths, this LP has an extra advantage: it can be solved quickly without using the ellipsoid algorithm. We note, however, that our method would yield the same approximation ratios with the LP of Dinitz and Krauthgamer [11] as well.

Directed Steiner Forest. Consider the DIRECTED STEINER FOREST (DSF) problem, a basic network design problem for directed graphs: given a directed graph $G = (V, E)$ with edge costs and a collection $D \subseteq V \times V$ of vertex pairs, find a minimum-cost subgraph of G that contains a path from u to v for every pair $(u, v) \in D$. DSF is an NP-hard problem and is known [12] to be quasi-NP-hard to approximate with ratio better than $2^{\log^{1-\epsilon} n}$ for all $\epsilon \in (0, 1)$. The best known approximation ratio for this problem is $O(n^{\epsilon} \cdot \min(n^{4/5}, m^{2/3}))$, due to Feldman, Kortsarz and Nutov [19]. Their algorithm has the same structure as the algorithms for DIRECTED k-SPANNER in [6] and [11]. Specifically, the LP relaxation that they formulate is closely related to that developed by Dinitz and Krauthgamer, if we replace edge costs by edge lengths. Our technique for the spanner problem also applies to the DSF problem, yielding an improved approximation ratio of $\tilde{O}(n^{2/3+\epsilon})$. We defer details to the full version.

2 An $\tilde{O}(\sqrt{n})$-Approximation for DIRECTED k-SPANNER

Our main result is stated in the following theorem.

Theorem 2.1. *There is a polynomial time randomized algorithm for* DIRECTED k-SPANNER *with expected approximation ratio* $O(\sqrt{n} \log n)$.

We present two algorithms to prove Theorem 2.1: an algorithm for the general case (whose description is completed in Section 2.2) and a simpler and more efficient algorithm for the special case when all edges have unit length (whose description is completed in Section 3).

Let $G = (V, E)$ be a directed graph with edge lengths $d : E \to \mathbb{R}^{\geq 0}$, given as input to our algorithm, and OPT be the size of its sparsest k-spanner. We assume that G is weakly connected. Otherwise, our algorithm should be executed for each weakly connected component separately.

Definition 2.1. *For an edge $(s, t) \in E$, let $G^{s,t} = (V^{s,t}, E^{s,t})$ be the subgraph of G induced by the vertices on paths from s to t of length at most $k \cdot d(s, t)$.*

Definition 2.2 (Thick and thin edges). *Let β be a parameter in $[1, n]$. If $|V^{s,t}| \geq n/\beta$, the corresponding edge (s, t) is* thick*, and otherwise, it is* thin*. The set of all thin edges is denoted by $\mathcal{E}$. In Sections 2.1–3, we shall always assume that $\beta = \sqrt{n}$.*

Our general strategy is to solve the problem separately for thick and thin edges. We find two sets of edges, E' and E'', such that for each edge $(s, t) \in E$, the required path of length at most $k \cdot d(s, t)$ from s to t is contained in E' if (s, t) is thick and in E'' if (s, t) is thin. The expected size of both sets is $O(\beta \log n \cdot OPT)$.

In Section 2.1, we describe how to obtain E' using random sampling. In Section 2.2, we describe how to obtain E'' in the general case, using randomized rounding of a fractional solution to an LP, thus completing the proof of Theorem 2.1. For graphs with unit edge lengths, the general method is the same, but we use a different LP (see Section 3).

2.1 Sampling

We say that an edge $(s, t) \in E$ is *settled* if the k-spanner property for this edge is satisfied, i.e., the selected set of edges contains a path of length at most $k \cdot d(s, t)$ from s to t. The following procedure uses random sampling to construct E'.

Algorithm 1 SAMPLE(β)

$E' \leftarrow \varnothing$, $S \leftarrow \varnothing$;
for $i = 1$ to $\beta \ln n$ **do**
  $v \leftarrow$ a uniformly random element of V;
  $T_v^{in} \leftarrow$ a shortest path in-arborescence rooted at v;
  $T_v^{out} \leftarrow$ a shortest path out-arborescence rooted at v;
  $E' \leftarrow E' \cup T_v^{in} \cup T_v^{out}$, $S \leftarrow S \cup \{v\}$; //Set S is used only in the analysis.
end for
Add all unsettled thick edges to E';
return E'.

Lemma 2.1. *Algorithm 1, in polynomial time, computes a set E' that settles all thick edges and has expected size at most $3\beta \ln n \cdot OPT$.*

Proof. After the execution of the **for**-loop in Algorithm 1, $|E'| \leq 2(n-1)\beta \ln n \leq 2\beta \ln n \cdot OPT$. The last inequality holds because $OPT \geq n-1$ for weakly connected graphs G.

If some vertex v from a set $V^{s,t}$ appears in the set S of vertices selected by Sample, then T_v^{in} and T_v^{out} contain shortest paths from s to v and from v to t, respectively. Thus, both paths are contained in E'. Since $v \in V^{s,t}$, the sum of lengths of these two paths is at most $k \cdot d(s,t)$. Therefore, if $S \cap V^{s,t} \neq \varnothing$, then the edge (s,t) is settled. For a thick edge (s,t), the set $S \cap V^{s,t}$ is empty with probability at most $(1-1/\beta)^{\beta \ln n} \leq e^{-\ln n} = 1/n$. Thus, the expected number of unsettled thick edges added to E' in Step 8 of Sample is at most $|E|/n \leq n-1 \leq OPT$.

Step 8 ensures that E', returned by the algorithm, settles all thick edges. Computing shortest path in- and out-arborescences and determining whether an edge is thick can be done in polynomial time. □

2.2 Antispanners, LP and the Separation Oracle

In this section, we introduce *antispanners*, a notion used in the description of our algorithm for Directed k-Spanner and essential in the analysis of all algorithms. It is needed in the parts of the algorithms that settle thin edges. Then, we formulate an LP relaxation of the problem of settling thin edges and present our approximation algorithm, proving Theorem 2.1.

Antispanners. For a given edge (s,t), we define an antispanner to be a subset of edges of G, such that if we remove this subset of edges from G, the length of the shortest path from s to t becomes larger than $k \cdot d(s,t)$.

Definition 2.3 (Antispanners). *A set $C \subseteq E$ is an antispanner for an edge $(s,t) \in E$ if $G' = (V, E \setminus C)$ contains no path from s to t of length at most $k \cdot d(s,t)$. If no proper subset of an antispanner C is an antispanner, we say that C is minimal.*

Thus, the edge set of a k-spanner of G must intersect all antispanners for all edges of G. In other words, it has to be a hitting set for all minimal antispanners.

We now prove that if a graph $(V, E' \cup E'')$ is not a k-spanner, then we can efficiently find a thin edge $(s,t) \in E$ and a minimal antispanner C that does not intersect E''.

Lemma 2.2. *There exists a polynomial time algorithm that, given a set of edges $E'' \subset E$ and a thin edge $(s,t) \in E$, outputs a minimal antispanner $C \subset E^{s,t} \setminus E''$ if there is no directed path from s to t of length at most $k \cdot d(s,t)$ in E''.*

Proof. The algorithm first checks if there exists a directed path from s to t of length at most $k \cdot d(s,t)$ in E''. If there is no such path then $E^{s,t} \setminus E''$ is an antispanner. (Note that all paths between s and t of lengths at most $k \cdot d(s,t)$ in G lie in the subgraph $G^{s,t} = (V^{s,t}, E^{s,t})$). The algorithm sets $C = E^{s,t} \setminus E''$ and then sequentially deletes all edges $(u,v) \in C$ such that $C \setminus \{(u,v)\}$ is an antispanner. When no more such edges are left, the algorithm returns C. □

$$\text{Minimize} \sum_{e \in E} x_e \text{ subject to:} \tag{1}$$

$$\sum_{e \in C} x_e \geq 1 \qquad \forall C \in \mathcal{S} \tag{2}$$

$$x_e \geq 0 \qquad \forall e \in E \tag{3}$$

Fig. 1. Linear program for the arbitrary-length case, LP-A

Linear Program. Since E' from Section 2.1 already settles the thick edges, our goal is to design a randomized procedure that finds a subset of edges $E'' \subset E$ that intersects all minimal antispanners for all thin edges. This condition can be expressed using linear program LP-A (see Fig. 1). This LP has a variable x_e for each edge $e \in E$ and a constraint (2) for each minimal antispanner C for thin edges. Set $\mathcal{S}$ is the set of all minimal antispanners for thin edges. In the integral solution $\{x_e^{int}\}$ corresponding to a k-spanner with edge set $E'' \subset E$, $x_e^{int} = 1$ if $e \in E''$ and $x_e^{int} = 0$ otherwise. All constraints (2) are satisfied for $\{x_e^{int}\}$ since E'' intersects every antispanner. The value of the objective function $\sum_e x_e^{int}$ equals the size of E''. Hence, the LP is a valid relaxation.

For ease of presentation, we assume that we have guessed OPT, the size of the optimal spanner. (We can try all values in $\{n-1, \ldots, n^2\}$ for OPT and output the best spanner found in all iterations). We replace the objective function (1) with

$$\sum_{e \in E} x_e \leq OPT. \tag{4}$$

Separation Oracle. Our LP has polynomially many variables and exponentially many constraints. We solve it using the ellipsoid algorithm with a separation oracle. Our separation oracle receives a fractional vector $\{x_e^*\}$ (satisfying (3), (4)) and outputs either a violated constraint (2) for some antispanner C or a set E'' of size at most $2OPT \cdot \sqrt{n} \ln n$ such that $E' \cup E''$ is the edge set of a k-spanner. Specifically, if $\{x_e^*\}$ is a feasible solution, then the separation oracle returns a set E''.

The separation oracle works as follows: it first samples a random set of edges E'' picking each $e \in E$ with probability $\min(x_e^* \sqrt{n} \ln n, 1)$:

Algorithm 2 RandomizedSelection(x_e^*)

1: $E'' \leftarrow \varnothing$;
2: **for** each edge $e \in E$ **do**
3: $\quad p_e \leftarrow \min(1, \sqrt{n} \ln n \cdot x_e^*)$;
4: $\quad$ Add e to E'' with probability p_e;
5: **end for**
6: **return** E''.

Then if $(V, E' \cup E'')$ is a spanner and $|E''| \leq 2OPT \cdot \sqrt{n} \ln n$, it outputs E''. If $|E''| > 2OPT \cdot \sqrt{n} \ln n$, the separation oracle fails. If $(V, E' \cup E'')$ is not a spanner, the algorithm finds a thin edge (s,t) such that there is no directed path of length $k \cdot d(s,t)$ from s to t in (V, E''), then it finds a minimal antispanner $C \subset E^{s,t} \setminus E''$ using a greedy algorithm (see Lemma 2.2 below for details; note that $E^{s,t} \setminus E''$ is an antispanner) and if $\sum_{e \in C} x_e < 1$, outputs this violated constraint. If $\sum_{e \in C} x_e \geq 1$, the separation oracle fails.

We now show that the probability that the separation oracle fails during an execution of the ellipsoid algorithm is small.

Theorem 2.2. *The probability that during an execution of the ellipsoid algorithm the separation oracle fails is exponentially small in n.*

Proof. As discussed above, there are two different events, which can cause the separation oracle to fail:

1. The size of the sampled set E'' is too large. The expected size of E'' is at most $\sqrt{n} \ln n \sum_{e \in E} x_e \leq OPT \cdot \sqrt{n} \ln n$. By the Chernoff bound, $\Pr(|E''| > 2OPT \cdot \sqrt{n} \ln n) \leq e^{-c \cdot OPT \cdot \sqrt{n} \ln n} = e^{-\Omega(n \cdot \sqrt{n} \ln n)}$. Thus, the probability that the separation oracle fails because $|E''| > 2OPT \cdot \sqrt{n} \ln n$ is exponentially small.
2. The minimal antispanner found by the oracle doesn't correspond to a violated constraint (see discussion below). We prove that the probability that the separation oracle fails because $\sum_{e \in C} x_e^* \geq 1$ is exponentially small in Lemma 2.3.

Lemma 2.3. *The probability that there exists an edge (s,t) and a minimal antispanner C for it such that $\sum_{e \in C} x_e^* \geq 1$, but $C \subset E^{s,t} \setminus E''$ is at most $|E| \cdot e^{-\frac{1}{2}\sqrt{n} \ln n}$.*

Proof. First, we bound the total number of minimal antispanners for thin edges.

Proposition 2.1. *If (s,t) is a thin edge, then there are at most $(n/\beta)^{n/\beta}$ minimal antispanners for (s,t). In particular, if $\beta = \sqrt{n}$, then there are at most $\sqrt{n}^{\sqrt{n}}$ minimal antispanners.*

Proof. Fix a thin edge (s,t) and consider an arbitrary minimal antispanner C for (s,t). Let A_C be the outward shortest path tree (arborescence) rooted at s in the graph $(V^{s,t}, E^{s,t} \setminus C)$. Denote by $f_{A_C}(u)$ the distance from s to u in the tree A_C. If there is no directed path from s to u in A_C, we let $f_{A_C}(u) = \infty$. We show that $C = \{(u,v) \in E^{s,t} : f_{A_C}(u) + d(u,v) < f_{A_C}(v)\}$, and, thus A_C uniquely determines C for a given thin edge (s,t). If $(u,v) \in C$, then, since C is a *minimal* antispanner, there exists a path from s to t of length at most $kd(s,t)$ in the graph $(V, E \setminus C \cup \{(u,v)\})$, this path must lie in $(V^{s,t}, E^{s,t} \setminus C \cup \{(u,v)\})$ and must contain the edge (u,v). Thus, the distance from s to t in the graph $(V^{s,t}, E^{s,t} \setminus C \cup \{(u,v)\})$ is at most $k \cdot d(s,t)$ and is strictly less than $f_{A_C}(t)$. Hence, A_C is not the shortest path tree in the graph $(V^{s,t}, E^{s,t} \setminus C \cup \{(u,v)\})$. Therefore, $f_{A_C}(u) + d(u,v) < f_{A_C}(v)$. If $(u,v) \in E^{s,t}$ satisfies the condition $f_{A_C}(u) + d(u,v) < f_{A_C}(v)$, then $(u,v) \notin E^{s,t} \setminus C$, otherwise A_C would not be the shortest path tree, hence $(u,v) \in C$.

We now count the number of outward trees rooted at s in $(V^{s,t}, E^{s,t} \setminus C)$. For every vertex $u \in V^{s,t}$ we may choose the parent vertex in at most $|V^{s,t}|$ possible ways (if a vertex is isolated we assume that it is its own parent), thus the total number of trees is at most $|V^{s,t}|^{|V^{s,t}|} \le (n/\beta)^{n/\beta}$. □

Proposition 2.2. *For an edge $(s,t) \in E$ and a minimal antispanner C for (s,t) satisfying $\sum_{e \in C} x^*_e \ge 1$, the probability that $E'' \cap C = \varnothing$ is at most $e^{-\sqrt{n}\ln n}$.*

Proof. Suppose there exists $(u,v) \in C$ such that $x^*_e \ge (\sqrt{n}\ln n)^{-1}$. In this case, $(u,v) \in E''$ with probability 1, and we are already done. Otherwise, for $(u,v) \in C$, the probability that $(u,v) \in E''$ is exactly $\sqrt{n}\ln n \cdot x_e$. The probability that no edges of C are in E'' is, therefore,

$$\prod_{e \in C}(1 - \sqrt{n}\ln n \cdot x^*_e) < \exp\left(-\sum_{e \in C}\sqrt{n}\ln n \cdot x^*_e\right) \le e^{-\sqrt{n}\ln n}.$$

The first inequality above follows from the fact that $1 - x < \exp(-x)$ for $x > 0$. The second one holds because $\sum_{e \in C} x^*_e \ge 1$. □

The proof of Lemma 2.3 is completed by using Proposition 2.2 and Proposition 2.1 and taking a union bound over all minimal antispanners for all thin edges. □

The proof of Theorem 2.2 is completed by using Lemma 2.3 and taking a union bound over all iterations of the ellipsoid algorithm, the number of which is bounded by a polynomial. □

Proof of Theorem 2.1.

Proof. The thick edges can be settled by running Sample($\sqrt{n}$), according to Lemma 2.1. The thin edges can be settled by running the ellipsoid algorithm as described above. The ellipsoid algorithm terminates in polynomial time. With exponentially small probability, we allow the separation oracle to fail (as shown in Theorem 2.2), in which case we output a spanner containing all edges E. Thus, the expected size of the set E'' is at most $2OPT \cdot \sqrt{n}\ln n + o(1)$ and the resulting approximation ratio of the algorithm is $O(\sqrt{n}\ln n)$.

3 LP and Rounding for Graphs with Unit-Length Edges

In this section, we describe how to settle the thin edges, and thus prove Theorem 2.1, for the case of unit-length edges. Our motivation for presenting this special case is two-fold. First, we show that for the unit-length case, one can directly formulate a polynomial-sized LP relaxation, and this makes the approximation algorithm more efficient. Second, the LP used here will be convenient in presenting the improved approximation for 3-spanners in Section 4.

In order to define and analyze the LP, we need to introduce some notation.

Definition 3.1 (Layered expansion). *Given a directed graph $G = (V, E)$, its* layered expansion *is a directed graph $\hat{G} = (\hat{V}, \hat{E})$, satisfying the following:*

1. *Let $\hat{V} = \{v_i : v \in V \text{ and } i \in \mathbb{Z}^{\geq 0}\}$, where v_i denotes the i-th copy of v. The set of all the i-th copies of nodes in V is the i-th layer of $\hat{V}$.*
2. *Let $L = \{(u, u) : u \in V\}$ be the set of loops. Define the i-th copy of an edge $e = (u, v)$ to be $e_i = (u_i, v_{i+1})$, and the i-th copy of a loop $e = (u, u)$ to be $e_i = (u_i, u_{i+1})$. Let $\hat{E} = \{e_i : e \in E \cup L \text{ and } i \in \mathbb{Z}^{\geq 0}\}$.*

We use layered expansion $\hat{G}$ to describe paths in G. Note that $\hat{G}$ contains a path from u_0 to v_ℓ if and only if G contains a path from u to v of length at most ℓ.

Recall that $\mathcal{E}$ denotes the set of thin edges. For $(s, t) \in \mathcal{E}$, we consider the subgraph of $\hat{G}$ consisting of all paths that can be used by a k-spanner:

Definition 3.2 (Edge network). *For an edge $(s, t) \in \mathcal{E}$ and $k \geq 1$, the* edge network *is a subgraph $\hat{G}_k^{s,t} = (\hat{V}_k^{s,t}, \hat{E}_k^{s,t})$ of $\hat{G}$ with a source $\bar{s} = s_0$ and a sink $\bar{t} = t_{k \cdot d(s,t)}$, such that $\hat{G}_k^{s,t}$ contains all nodes and edges on paths from $\bar{s}$ to $\bar{t}$.*

Now, consider the linear program LP-U defined in Figure 2 below. LP-U has variables of two types: x_e, where $e \in E$, and $f_{e_i}^{s,t}$, where $(s, t) \in \mathcal{E}$ and $e_i \in \hat{E}_k^{s,t}$. A variable x_e represents whether the edge e is included in the k-spanner. A variable $f_{e_i}^{s,t}$ represents flow along the edge e_i in $\hat{G}_k^{s,t}$ (integer flow in $\hat{G}_k^{s,t}$ with value 1 is simply a path of length at most k). We denote the sets of incoming and outgoing edges for a vertex $v_i \in \hat{G}_k^{s,t}$ by $In(v_i)$ and $Out(v_i)$, respectively.

Minimize $\sum_{e \in E} x_e$ subject to:

Flow requirement	$\sum_{e_0 \in Out(s_0)} f_{e_0}^{s,t} \geq 1$	$\forall (s,t) \in \mathcal{E}$
Flow conservation	$\sum_{e_{i-1} \in In(v_i)} f_{e_{i-1}}^{s,t} - \sum_{e_i \in Out(v_i)} f_{e_i}^{s,t} = 0$	$\forall (s,t) \in \mathcal{E}, \forall v_i \in \hat{V}_k^{s,t} \setminus \{\bar{s}, \bar{t}\}$
Capacity constraints	$x_e - \sum_{i=0}^{k-1} f_{e_i}^{s,t} \geq 0$	$\forall (s,t) \in \mathcal{E}, \forall e \in E$
	$x_e \geq 0$	$\forall e \in E$
	$f_{e_i}^{s,t} \geq 0$	$\forall (s,t) \in \mathcal{E}, \forall e_i \in \hat{E}_k^{s,t}$

Fig. 2. Linear program for the unit-length case, LP-U

Note that to write down LP-U, we only need to know V, E, k and the set of thin edges, $\mathcal{E}$. The first three are inputs to the algorithm, and $\mathcal{E}$ can be computed in polynomial time. LP-U can be written down and solved in polynomial time because it has $O(|E|^2 \cdot k) = O(n^5)$ variables and constraints.[4] Thus, unlike the

[4] More precisely, LP-U has $O(|\mathcal{E}| \times |V^{s,t}|^3) = O(n^{3.5})$ variables and constraints.

case of arbitrary lengths, one does not need to invoke the ellipsoid algorithm here.

Given x^*_e, the fractional solution of LP-U, we construct the set E'' by first running Algorithm 2 and then adding all unsettled thin edges. Because sets of fractional solutions x^*_e to LP-U and LP-A are equal, one can show that the set $E' \cup E''$ forms a k-spanner with high probability and the size of this spanner is $O(OPT \cdot \sqrt{n} \log n)$. We give a direct proof of this fact in the full version.

4 An $\tilde{O}(n^{1/3})$-Approximation for Directed 3-Spanner with Unit-Length Edges

In this section, we show an improved approximation for the special case of Directed 3-Spanner with unit length edges. At a high level, our analysis is a combination of the technique we described for Directed k-Spanner with the technique of Dinitz and Krauthgamer [11] for Directed 3-Spanner. The algorithm for Directed 3-Spanner in [11] does not use sampling, which makes their result applicable to the problem on graphs with arbitrary edge *cost*, where the total edge cost is minimized rather than the total number of edges in the spanner. By combining with sampling, we can improve the approximation ratio for graphs with unit edge costs and lengths.

Theorem 4.1. *There is a polynomial time randomized algorithm for* Directed 3-Spanner *for graphs with unit edge lengths with expected approximation ratio* $O(n^{1/3} \log^2 n)$.

Proof. We define thick and thin edges as in Definition 2.2, with $\beta = n^{1/3}$, and we run Sample($n^{1/3}$). By Lemma 2.1, this settles all thick edges with edge set E' that on the average has size at most $3n^{1/3} \ln n \cdot OPT$. Then we obtain solution x^* of the linear program LP-U from Fig. 2 and use randomized rounding to obtain edge set E'' that settles all thin edges with high probability. However, we need to use a different method of rounding that takes advantage of the fact that our spanners provides paths of length 3. We could have used Algorithm 2 from [11] with $\rho = \tilde{\Theta}(n^{1/3})$, but instead we give a simplified rounding scheme.

Algorithm 3 RandomizedSpannerSelection(x^*_e)

1: $E'' \leftarrow \varnothing$;
2: **for** each vertex $u \in V$ **do**
3: Let r_u be chosen i.i.d. uniformly from $[0, 1]$;
4: **end for**
5: **for** each edge $e = (u, v) \in E$ **do**
6: Add e to E'' if $r_u r_v \le x^*_{u,v} \alpha\, n^{1/3} \ln n$; //$\alpha$ is a constant less than 10
7: **end for**
8: **return** E''.

It suffices to prove the following two lemmas. Lemma 4.1 bounds the expected size of E''. Lemma 4.2 shows that E'' settles almost all thin edges

Lemma 4.1 (analog of Lemma 4.1 in [11]). $\mathbb{E}[|E''|] = O(OPT n^{1/3} \ln^2 n)$.

Lemma 4.2 (analog of Lemma 4.2 in [11]). *If (s,t) is a thin edge, E'' contains a path from s to t of length at most 3 with probability at least $1 - 1/n$.*

By this lemma, the expected number of unsettled thin edges is at most $|E|/n \leq n \leq OPT$, so one can simply add the unsettled edges to the solution.

It remains to prove Lemmas 4.1 and 4.2. In the proof of Lemma 4.1, we use the following fact whose proof we omit for space considerations:

Lemma 4.3. *If $q \leq 1$, $\mathbf{P}r[r_u r_v \leq q] = q(1 - \ln q)$.*

Proof (of Lemma 4.1). Let $A = \{e \in E : x_e^* \alpha n^{1/3} \ln n \geq 1/n\}$ and $B = E \setminus A$. We use two estimates for OPT: $OPT_1 = \sum_e x_e^*$ and $OPT_2 = |B|/n$. Clearly,

$\mathbb{E}[|E''|] = \mathbb{E}[|E'' \cap A|] + \mathbb{E}[|E'' \cap B|]$;

$\mathbb{E}[|E'' \cap A|] \leq OPT_1 \times \alpha n^{1/3} \ln n (1 + \ln n)$;

$\mathbb{E}[|E'' \cap B|] \leq OPT_2 \times (1 + \ln n)$.

Both inequalities follow from Lemma 4.3. □

We defer the proof of Lemma 4.2 to the full version.

□

5 Conclusion

We gave approximation algorithms with ratio $\tilde{O}(\sqrt{n})$ for Directed k-Spanner and with ratio $\tilde{O}(n^{1/3})$ for Directed 3-Spanner with unit length edges. It remains an interesting open question whether one improve the approximation ratio to $\tilde{O}(n^{1/3})$ for arbitrary lengths and larger k, thus matching the integrality gap shown by Dinitz and Krauthgamer. Our algorithm for Directed k-Spanner applies to the k-Transitive-Closure Spanner problem [6], which can be reformulated as a special case of Directed k-Spanner. It also straightforwardly extends to the Client-Server k-Spanner problem and the k-Diameter Spanning Subgraph problem [16].

References

1. Althöfer, I., Das, G., Dobkin, D., Joseph, D., Soares, J.: On sparse spanners of weighted graphs. Discrete & Computational Geometry 9(1), 81–100 (1993)
2. Awerbuch, B.: Communication-time trade-offs in network synchronization. In: PODC. pp. 272–276 (1985)
3. Baswana, S., Sen, S.: Approximate distance oracles for unweighted graphs in expected $\tilde{O}(n^2)$ time. ACM Transactions on Algorithms 2(4), 557–577 (2006)
4. Berman, P., Raskhodnikova, S., Ruan, G.: Finding sparser directed spanners. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS. LIPIcs, vol. 8, pp. 424–435. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2010)

5. Bhattacharyya, A., Grigorescu, E., Jha, M., Jung, K., Raskhodnikova, S., Woodruff, D.P.: Lower bounds for local monotonicity reconstruction from transitive-closure spanners. In: Serna, M.J., Shaltiel, R., Jansen, K., Rolim, J.D.P. (eds.) APPROX-RANDOM. LNCS, vol. 6302, pp. 448–461. Springer (2010)
6. Bhattacharyya, A., Grigorescu, E., Jung, K., Raskhodnikova, S., Woodruff, D.: Transitive-closure spanners. In: SODA. pp. 932–941 (2009)
7. Cohen, E.: Fast algorithms for constructing t-spanners and paths with stretch t. SIAM J. Comput. 28(1), 210–236 (1998)
8. Cohen, E.: Polylog-time and near-linear work approximation scheme for undirected shortest paths. JACM 47(1), 132–166 (2000)
9. Cowen, L.: Compact routing with minimum stretch. J. Algorithms 38(1), 170–183 (2001)
10. Cowen, L., Wagner, C.G.: Compact roundtrip routing in directed networks. J. Algorithms 50(1), 79–95 (2004)
11. Dinitz, M., Krauthgamer, R.: Directed spanners via flow-based linear programs. In: Vadhan, S. (ed.) STOC. ACM (2011), to appear
12. Dodis, Y., Khanna, S.: Designing networks with bounded pairwise distance. In: STOC. pp. 750–759 (1999)
13. Elkin, M.: Computing almost shortest paths. In: PODC. pp. 53–62 (2001)
14. Elkin, M., Peleg, D.: Strong inapproximability of the basic k-spanner problem. In: ICALP. pp. 636–647 (2000)
15. Elkin, M., Peleg, D.: The client-server 2-spanner problem with applications to network design. In: SIROCCO. pp. 117–132 (2001)
16. Elkin, M., Peleg, D.: Approximating k-spanner problems for $k \geq 2$. Theor. Comput. Sci. 337(1-3), 249–277 (2005)
17. Elkin, M., Peleg, D.: The hardness of approximating spanner problems. Theory Comput. Syst. 41(4), 691–729 (2007)
18. Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., Zhang, J.: Graph distances in the data-stream model. SIAM J. Comput. 38(5), 1709–1727 (2008)
19. Feldman, M., Kortsarz, G., Nutov, Z.: Improved approximating algorithms for Directed Steiner Forest. In: Mathieu, C. (ed.) SODA. pp. 922–931. SIAM (2009)
20. Jha, M., Raskhodnikova, S.: Testing and reconstruction of Lipschitz functions with applications to data privacy. Electronic Colloquium on Computational Complexity (ECCC) TR11-057 (2011)
21. Kortsarz, G., Peleg, D.: Generating sparse 2-spanners. J. Algorithms 17(2), 222–236 (1994)
22. Kortsarz, G.: On the hardness of approximating spanners. Algorithmica 30(3), 432–450 (2001)
23. Peleg, D., Schäffer, A.A.: Graph spanners. J. Graph Theory 13(1), 99–116 (1989)
24. Peleg, D., Ullman, J.D.: An optimal synchronizer for the hypercube. SIAM J. Comput. 18(4), 740–747 (1989)
25. Peleg, D., Upfal, E.: A trade-off between space and efficiency for routing tables. JACM 36(3), 510–530 (1989)
26. Raskhodnikova, S.: Transitive-closure spanners: a survey. In: Goldreich, O. (ed.) Property Testing. vol. LNCS 6390, pp. 167–196. Springer, Heidelberg (2010)
27. Roditty, L., Thorup, M., Zwick, U.: Roundtrip spanners and roundtrip routing in directed graphs. ACM Transactions on Algorithms 4(3) (2008)
28. Thorup, M., Zwick, U.: Compact routing schemes. In: SPAA. pp. 1–10. ACM (2001)
29. Thorup, M., Zwick, U.: Approximate distance oracles. JACM 52(1), 1–24 (2005)