

Directed Spanners via Flow-Based Linear Programs*

Michael Dinitz
Weizmann Institute of Science
michael.dinitz@weizmann.ac.il

Robert Krauthgamer
Weizmann Institute of Science
robert.krauthgamer@weizmann.ac.il

ABSTRACT

We examine directed spanners through flow-based linear programming relaxations. We design an $\tilde{O}(n^{2/3})$ -approximation algorithm for the directed k -spanner problem that works for all $k \geq 1$, which is the first sublinear approximation for arbitrary edge-lengths. Even in the more restricted setting of unit edge-lengths, our algorithm improves over the previous $\tilde{O}(n^{1-1/k})$ approximation [BGJ⁺09] when $k \geq 4$. For the special case of $k = 3$ we design a different algorithm achieving an $\tilde{O}(\sqrt{n})$ -approximation, improving the previous $\tilde{O}(n^{2/3})$ [EP05, BGJ⁺09] (independently of our work, an $\tilde{O}(n^{1-1/\lceil k/2 \rceil})$ was recently devised [BRR10]). Both of our algorithms easily extend to the *fault-tolerant* setting, which has recently attracted attention but not from an approximation viewpoint. We also prove a nearly matching integrality gap of $\Omega(n^{1/3-\epsilon})$ for every constant $\epsilon > 0$.

A virtue of all our algorithms is that they are relatively simple. Technically, we introduce a new yet natural flow-based relaxation, and show how to approximately solve it even when its size is not polynomial. The main challenge is to design a rounding scheme that “coordinates” the choices of flow-paths between the many demand pairs while using few edges overall. We achieve this, roughly speaking, by randomization at the level of *vertices*.

Categories and Subject Descriptors

G.2.2 [Discrete Mathematics]: Graph Theory

General Terms

Algorithms, Theory

Keywords

Approximation Algorithms, Spanners, Linear Programming

*A full version appears at <http://arxiv.org/abs/1011.3701>. Work supported in part by The Israel Science Foundation (grant #452/08), and by a Minerva grant.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

STOC'11, June 6–8, 2011, San Jose, California, USA.

Copyright 2011 ACM 978-1-4503-0691-1/11/06 ...\$10.00.

1. INTRODUCTION

We examine several directed spanner problems from the perspective of approximation via a linear programming (LP) relaxation. In particular, we design flow-based LP relaxations for these classical NP-hard problems, and then investigate how well these relaxations approximate the optimal spanner, providing nearly matching upper and lower bounds. We begin by introducing the spanner problems that we consider, focusing throughout on *directed* graphs; we briefly compare to undirected graphs in Section 1.4.

1.1 Spanner Problems

Let $G = (V, E)$ be a strongly connected directed graph.¹ A k -spanner of G , for $k \geq 1$, is a subgraph $G' = (V, E')$, that preserves all pairwise distances within factor k , i.e. for all $u, v \in V$,

$$d_{G'}(u, v) \leq k \cdot d_G(u, v). \quad (1)$$

Here and throughout, d_H denotes the shortest-path distance in a graph H . It is easy to see that requiring (1) only for edges $(u, v) \in E$ suffices.

In the directed k -spanner problem with unit edge-lengths, the input is the graph G and the goal is to find a k -spanner G' having the minimum number of edges. We allow the *stretch* k to be a function of $n = |V|$, e.g. $k = O(\log n)$, and in fact some of our results are most interesting when $k = \Omega(\log n)$. This definition was introduced by Peleg and Schäfer [PS89] (in particular, they showed the problem is NP-hard), and since then it has been studied extensively, with applications ranging from routing in networks (e.g. [AP95, TZ05]) to solving linear systems (e.g. [ST04, EEST08]).

The above definition has several natural generalizations. An obvious one is to let G have nonnegative edge-lengths, leading to more complicated distances. This is the directed k -spanner problem *with arbitrary edge-lengths*. Another generalization, introduced in [CLPR09], incorporates fault tolerance: a k -spanner G' is r -vertex-tolerant if for all $F \subseteq V$ with $|F| \leq r$ we have that $G' \setminus F$ is a k -spanner of $G \setminus F$. The definition of r -edge-tolerant is the same, except that F is a subset of E rather than of V . Clearly, the special case $r = 0$ is just the standard notion defined above. This paper address both of these generalizations.

Yet another generalization of the problem is the client-server model [EP01]: the input contains also a set $C \subseteq E$ of so-called client edges and a set $S \subseteq E$ of server edges,

¹The assumption of strong connectivity is for notational convenience, although the definitions and all our results extend easily to all digraphs.

the requirement (1) is only needed for edges in \mathcal{C} , while the spanner is only allowed to use edges in \mathcal{S} (i.e. $E' \subseteq \mathcal{S}$). Obviously, the case $\mathcal{C} = \mathcal{S} = E$ is just the standard notion defined above. Our results extend to this model in a straightforward manner, but for the sake of exposition we shall not address it directly.

1.2 Results

We first present a flow-based LP relaxation for spanner problems (Section 2). This relaxation is quite natural but appears to be new, and in particular it differs from the ones used in [DK99, BGJ⁺09]. We then use this LP relaxation to obtain the approximation algorithms described below (see also Table 1).

General stretch k .

Our first algorithmic result is an $\tilde{O}(n^{2/3})$ -approximation for the directed k -spanner problem that works for all $k \geq 1$, even with arbitrary edge-lengths (Section 3). This is the first approximation algorithm that handles the more general case of arbitrary edge-lengths. Furthermore, using the reduction of [BGJ⁺09] from Transitive-Closure k -spanner to directed k -spanner we obtain for the former problem an $\tilde{O}(n^{2/3})$ -approximation, improving over their $\tilde{O}(\min\{n^{1-1/k}, n/k^2\})$ -approximation for all $k \ll n^{1/6}$.

Previous upper bounds for this problem worked only in the case when all edge lengths are unit. The most relevant result is due to Bhattacharyya, Grigorescu, Jung, Raskhodnikova, and Woodruff [BGJ⁺09], who gave an $\tilde{O}(n^{1-1/k})$ -approximation. When $k \geq 4$ our result gives a polynomial improvement. Recently, and independently of our work, Berman, Raskhodnikova, and Ruan [BRR10] improved the result of [BGJ⁺09] by designing an algorithm with approximation ratio $\tilde{O}(n^{1-1/\lceil k/2 \rceil})$. For all $k \geq 7$ our result gives an improvement over this new bound. Perhaps more importantly, we show that the approximation need not increase with k and provide the first sublinear (in n) approximation ratio for $k \geq \log n$, as well as providing the first nontrivial upper bound for arbitrary edge lengths.

We complement the above algorithmic result by showing that our (rather natural) LP relaxation has an integrality gap of $\tilde{\Omega}(\frac{1}{k}n^{1/3-\epsilon})$ for every constant $\epsilon > 0$, even in the unit-length case (Section 3.2). Previously, Elkin and Peleg [EP07] proved that for every fixed $0 < \epsilon, \delta < 1$ and $3 \leq k = o(n^\delta)$, approximating the directed k -spanner with unit edge-lengths problem within ratio $2^{\log^{1-\epsilon} n}$ is quasi-NP-hard (similar hardness results were already known for smaller ranges of k [Kor01]). We conclude that a polynomial approximation (independent of k) is probably the best one can hope for, and specifically the best possible exponent appears to be in the range $[1/3, 2/3]$.

Stretch $k = 3$.

For directed 3-spanner with unit edge-lengths we achieve an even better $\tilde{O}(\sqrt{n})$ -approximation (Section 4). Notice that this approximation factor matches, up to lower order factors, the $O(\sqrt{n})$ -approximation known for undirected graphs (an immediate consequence of the absolute guarantee of [ADD⁺93] that every undirected graph has a k -spanner with $O(n^{1+2/(k+1)})$ edges). The previous approximation known for this case is $\tilde{O}(n^{2/3})$, first proved by Elkin and Peleg [EP05]. A similar $\tilde{O}(n^{2/3})$ -approximation can be ob-

tained by the aforementioned algorithm of [BGJ⁺09], as well as by our algorithm for general k . As mentioned above, an $\tilde{O}(\sqrt{n})$ -approximation for this case (as well as for the case of $k = 4$) was recently provided by Berman, Raskhodnikova, and Ruan [BRR10] independently of our work. However, since our techniques are based on rounding a linear program (as opposed to the combinatorial techniques of [BRR10]) our algorithm extends easily to the setting where each edge e has some nonnegative cost c_e and the goal is to find a minimum cost 3-spanner. Moreover, our performance guarantee is relative to the value of the linear programming relaxation rather than the integral optimum, which is a slightly stronger guarantee.

Stretch $k = 2$.

This case (directed 2-spanner with unit-length edges) is rather exceptional and is known to have tight approximation bounds: $O(\log n)$ approximation [KP94, EP01] and $\Omega(\log n)$ NP-hardness [Kor01].² We show similar bounds on the integrality gap of our LP relaxation (in Section 5), a finding that is not very surprising but affirms the strong connection between our LP relaxation and the approximability threshold.

Fault-tolerant spanners.

We also adapt our algorithms to the fault-tolerant setting, albeit restricted to unit edge-lengths (see Sections 3.1 and 4.1). The fault-tolerant setting is significantly more complicated: the LP relaxation might have an exponential number of both variables and constraints (see Section 2.1), and we must resort to bicriteria approximations when the number of faults r is not constant. Generally speaking, the approximation factors we obtain grow with r like k^r in the first algorithm (for general k), and linearly in r in the second algorithm (for $k = 3$ with unit-length edges). These are the first results for fault-tolerant spanners in *directed* graphs. For undirected graphs, absolute bounds (i.e. not as approximation factors) are known [CLPR09, DK11].

1.3 Techniques

All of our approximation algorithms rely on solving the LP relaxation and rounding the resulting “fractional” solution. In some cases the LP relaxation does not have polynomial size, but we can solve it within a reasonable approximation in polynomial time by reducing it (via duality and the ellipsoid algorithm) to a problem known in the literature as Restricted Shortest Path (Theorems 2.1 and 2.2). A virtue of all our algorithms is that they are relatively simple, and thus can be extended to more complicated scenarios with little effort, as is evident in the fault-tolerance case.

Our main technical contribution is to design new rounding procedures which use few edges but are effective in creating many suitable paths. Two very natural and well-known rounding techniques fail miserably: (1) rounding separately each edge proportional to its LP value (deterministically or randomly) is unlikely to form suitable paths, and (2) rounding separately each flow-path (say randomly à la Raghavan and Thompson [RT87]) will use far too many edges. The challenge is thus to “coordinate” the selection of edges so that they tend to create suitable paths. Put differently, each

²It is possible to refine the approximation in terms of the graph’s average/maximum degree.

Directed k -Spanner with unit edge-lengths

Stretch	Our Approximation	Previous/Independent Approximations	Integrality gap
$k \geq 4$	$\tilde{O}(n^{2/3})$ Thm 3.3	$\tilde{O}(n^{1-1/k})$ [BGJ ⁺ 09] / $\tilde{O}(n^{1-1/\lceil k/2 \rceil})$ [BRR10]	$\Omega(\frac{1}{k} \cdot n^{1/3-\varepsilon})$ Thm 3.6
$k = 3$	$\tilde{O}(n^{1/2})$ Thm 4.3	$\tilde{O}(n^{2/3})$ [EP05, BGJ ⁺ 09] / $\tilde{O}(n^{1/2})$ [BRR10]	$\Omega(n^{1/3-\varepsilon})$ Thm 3.6
$k = 2$	$O(\log n)$ Thm 5.2	$O(\log n)$ [KP94, EP01]	$\Omega(\log n)$ Thm 5.1

Similarly but with r (vertex/edge) fault-tolerance

Stretch	Our Approximation	Previous Approximation
$k \geq 4$	$(\frac{1}{k(1+\varepsilon)}, O((\frac{(1+\varepsilon)r(k+r)^{k+r}n \ln n}{\varepsilon r^r k^k})^{2/3}))$ Thm 3.4	—
$k = 3$	$(\frac{1}{3(1+\varepsilon)}, \tilde{O}(rn^{1/2}))$ Thm 4.4	—
$k = 2$	$O(r \log n)$ Thm 5.3	—

Table 1: Summary of our approximation results for the case of unit edge-lengths

$(u, v) \in E$ can be seen as a demand pair with its own flow, and we need to select one flow-path for each pair in a way that is “biased” towards using the same edges. Ideally, we would like to select both the edges and the flow-paths proportionally to their LP value.

Our first algorithm, for general k , is based on classifying demand pairs according to whether they have “few” or “many” low-stretch paths. The key insight is to make this classification rely on counting *vertices* participating in low-stretch paths. The algorithm is then almost straightforward: applying a threshold rounding of the LP handles pairs of the first type, and building shortest-path arborescences from a small number of randomly chosen vertices handles pairs of the second type. This algorithm is described in Section 3. This basic idea of trading off an LP with random arborescences was used in previous algorithms [BGJ⁺09], but with a weaker linear program that limited the achievable approximation and required unit edge lengths.

To extend this algorithm to the fault-tolerant case we use the appropriate LP relaxation and apply the above rounding technique to several “perturbations” of the instance, each obtained by deleting from the graph a random subset of vertices/edges. This algorithm is described in Section 3.1. We believe that this perturbation technique, which we call failure sampling, is of independent interest, and may find future applications in related fault-tolerant problems. For example, it was recently shown [DK11] that a simple application of failure sampling combined with existing *undirected* spanner constructions gives a fault-tolerant undirected spanner construction with an absolute bound on its size that is only $\tilde{O}(r^2)$ larger than for a non-fault-tolerant spanner (where r is the number of faults), while the best previous construction [CLPR09] has size $\tilde{O}(k^r)$ larger than a non-fault-tolerant spanner.

Our second algorithm, for directed 3-spanner with unit edge-lengths, uses randomized rounding but at the level of *vertices* rather than edges or flow-paths. For every $v \in V$ we choose a random threshold $T_v \in [0, 1]$, and include in the solution every edge $(u, v) \in E$ for which, compared to the edge’s LP value, either $\min\{T_u, T_v\}$ is “small” or $\max\{T_u, T_v\}$ is “moderate”. The probability of including an edge in this solution is proportional to the edge’s LP value, but our conditions encourage positive correlation along a path (e.g. for edges sharing an endpoint). The proof boils down to considering a given demand pair, and analyzing the possibly many different flow-paths between them. In some

cases, we control the correlation between these flow-paths using Janson’s inequality (see e.g. [AS00, DP09]). But in other cases such correlation analysis is not effective, so we structurally “decompose” the paths into their first, second and third hops and then use standard concentration bounds separately for each hop plus some global arguments based on flow conservation. This rounding procedure extends to the fault-tolerant case very easily; we just repeat the rounding procedure several times with fresh coins. These algorithms are described in Section 4.

1.4 Concluding Remarks and Future Work

Our results for directed k -spanner with unit edge-lengths address what Elkin and Peleg [EP05] highlighted as two “challenging directions”: obtaining sublinear approximations for general k and improving over their $\tilde{O}(n^{2/3})$ approximation for $k = 3$. LP-based approaches are quite generic yet often times optimal, and thus it would not be surprising if the integrality gap of our LP relaxation gives away the problem’s true approximability threshold, which appears to be polynomial with exponent in the range $[1/3, 2/3]$ for all $k \geq 3$.

A bolder conjecture would be that this LP relaxation also exposes the approximability threshold for other spanner problems. One such family of problems is the undirected setting, whose consideration we defer to future work but briefly discuss some preliminary results. For $k = 3$ the known approximation (and thus integrality gap) is $O(\sqrt{n})$ by the absolute guarantee of [ADD⁺93] (or our Theorem 4.3), while our best integrality gap is $\tilde{\Omega}(n^{1/8})$, which we prove by a somewhat tricky analysis of the random graph $G_{n,p}$ for $p \approx 1/\sqrt{n}$. For larger k , the known approximation (and thus integrality gap) is $O(n^{2/(k+1)})$ via [ADD⁺93], while our best integrality gap is $n^{\Omega(1/k)}$, using the reduction designed by [EP07] from Min-Rep with large girth (which they conjecture to be NP-hard). Compared to [EP07], our integrality gap may be viewed as yet another indication of the inapproximability of the basic k -spanner problem (namely, undirected and with unit edge lengths), but with a factor much closer to the known approximation algorithm. One intriguing possibility, which is not ruled out by our integrality gaps or by the known hardness results, is that new rounding techniques for the linear program could give new *bicriteria* approximations for various spanner problems. It might be possible to lose a little on the required stretch, but gain enormously in the approximation of the size.

2. FLOW-BASED LP RELAXATION

We begin by describing the linear programming relaxation of the directed k -spanner problem that we will use. Consider an instance of the directed k -spanner problem: a directed graph $G = (V, E)$ and an assignment of lengths to the edges $d : E \rightarrow \mathbb{R}^+$. It is obviously sufficient to require stretch k only for edges in G , as this will imply a stretch of at most k for all other pairs. For $(u, v) \in E$, let $\mathcal{P}_{u,v}$ denote the set of all stretch k paths (in G) from u to v , i.e. valid paths whose length is within factor k of the shortest. It is easy to see that the following LP is a relaxation of the k -spanner problem. The variables are x_e , representing whether edge $e \in E$ is included in G' , and f_P , representing flow along path P .

$$\begin{aligned} \min \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_{u,v}: e \in P} f_P \leq x_e \quad \forall (u, v) \in E, \forall e \in E \\ & \sum_{P \in \mathcal{P}_{u,v}} f_P \geq 1 \quad \forall (u, v) \in E \\ & x_e \geq 0 \quad \forall e \in E \\ & f_P \geq 0 \quad \forall (u, v) \in E, \forall P \in \mathcal{P}_{u,v} \end{aligned} \tag{2}$$

In general LP (2) has exponential size. If the number of paths in $\mathcal{P}_{u,v}$ is at most polynomial for all $(u, v) \in E$ (for example if all edge lengths are unit and k is a constant) then the LP obviously has only a polynomial number of variables and constraints, so can be solved optimally. But in general we will need a different approach. We will work with the dual, which has a polynomial number of variables and an exponential number of constraints. So it is sufficient to find a separation oracle for the dual. It turns out that the separation problem for the dual is the *Restricted Shortest Path* problem, sometimes also called the *Length Constrained Lightest Path* problem. A PTAS is known for this problem [LR01, Has92], so we can approximately separate for the dual. This is enough to approximately solve the primal.

THEOREM 2.1. *For every $\epsilon > 0$ there is an algorithm that in polynomial time computes a $(1 + \epsilon)$ approximation to the optimal solution of LP (2).*

PROOF. The dual of LP (2) is LP (3), which has a variable for every edge and a variable for every pair of edges. The intuition is that the $y_{u,v}^e$ variables must form a “fractional cut” of (u, v) relative to stretch- k paths.

$$\begin{aligned} \max \quad & \sum_{(u,v) \in E} z_{u,v} \\ \text{s.t.} \quad & \sum_{(u,v) \in E} y_{u,v}^e \leq 1 \quad \forall e \in E \\ & z_{u,v} - \sum_{e \in P} y_{u,v}^e \leq 0 \quad \forall (u, v) \in E, \forall P \in \mathcal{P}_{u,v} \\ & z_{u,v} \geq 0 \quad \forall (u, v) \in E \\ & y_{u,v}^e \geq 0 \quad \forall (u, v) \in E, \forall e \in E \end{aligned} \tag{3}$$

To construct a separation oracle for this LP, note there are only a polynomial number ($|E|$) of constraints of the first type, so we can just check them one by one. For constraints of the second type, note that for every $u, v \in V$ the values $\{y_{u,v}^e\}_{e \in E}$ are just a non-negative edge-weighting, and the constraint just requires all of the original stretch k paths

to have total length (under this new weighting) of at least $z_{u,v}$. So we get the following problem: given two weightings of the same graph, find the shortest path under the second weighting subject to having length at most $T > 0$ under the first weighting (for some threshold T). If we could solve this we would have a separation oracle for the dual. Note that in the unit edge-length case stretch- k paths correspond exactly to paths with k or fewer hops, so we can solve this problem exactly using Bellman-Ford.

For the general lengths setting, this problem has been considered in the literature under the names “Length Constrained Lightest Path” and “Restricted Shortest Path”. An FPTAS is known to exist [LR01, Has92], which gives us an approximate separation oracle. So by using Ellipsoid with this oracle we find a polynomial number of constraints such that the optimal solution violates all of the other constraints (which we did not include) by at most $1 - \epsilon$, i.e. there might be paths where $(1 - \epsilon)z_{u,v} \leq \sum_{e \in P} y_{u,v}^e$. So if we simply let $z_{u,v} = (1 - \epsilon)z_{u,v}$ we have a feasible solution for LP (3) that is within $1 - \epsilon$ of optimal. So the optimum of LP (3) is at least $1 - \epsilon$ times the optimum of the compact dual (informally, changing to only a polynomial number of constraints did not affect the value of the optimal solution very much). Thus by strong duality if we solve a compact version of LP (2) that has only the variables corresponding to the constraints found by Ellipsoid on the dual (of which there are only a polynomial number) we get a solution of value equal to the optimum of the compact dual, which is at most $1/(1 - \epsilon)$ times the value of the actual dual (LP (3)). \square

2.1 Fault Tolerant Relaxation

There are two versions of the r -fault tolerant k -spanner problem, depending on whether we protect against edge faults or vertex faults. The idea in both cases is the same, so we shall focus on vertex faults: construct a subgraph H of G such that for every set F of at most r faulting vertices, $H \setminus F$ is a k -spanner of $G \setminus F$. We can change LP (2) to support this version by allowing a different set of flows $\{f_P^F\}$ for every such F , but using the same capacity variables $\{x_e\}$. More formally, let $\mathcal{P}_{u,v}^F$ be the set of stretch- k paths from u to v in $G \setminus F$. For $F \subseteq V$, let $E_F \subseteq E$ be the set of edges with neither endpoint in F (i.e. E_F is the set of edges which survive when the vertices in F are removed). Let $\mathcal{F} = \{F \subseteq V : |F| \leq r\}$ be the collection of possible fault sets. We will use the following relaxation for the vertex version (the edge version is analogous):

$$\begin{aligned} \min \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_{u,v}^F: e \in P} f_P^F \leq x_e \quad \forall F \in \mathcal{F}, \forall (u, v) \in E_F, \\ & \quad \quad \quad \forall e \in E_F \\ & \sum_{P \in \mathcal{P}_{u,v}^F} f_P^F \geq 1 \quad \forall F \in \mathcal{F}, \forall (u, v) \in E_F \\ & x_e \geq 0 \quad \forall e \in E \\ & f_P^F \geq 0 \quad \forall F \in \mathcal{F}, \forall (u, v) \in E_F, \\ & \quad \quad \quad \forall P \in \mathcal{P}_{u,v}^F \end{aligned} \tag{4}$$

It is easy to see that LP (4) has $n^{O(r)}$ constraints. Each possible fault set acts like an instance of the original spanner LP (2), except for the sharing of the capacity variables $\{x_e\}_{e \in E}$. So when we take the dual we get a program with

$n^{O(r)}$ variables, and if r is constant the separation oracle we designed for the non-fault-tolerant version suffices to separate this LP as well. When r is not constant this technique does not work as the dual will have a super-polynomial number of variables. Instead we will give a bicriteria algorithm, which in the unit-length case will find a fractional assignment to the x_e variables of cost at most $\frac{1+\epsilon}{\epsilon}$ times larger than the cost of the best fractional solution, and that supports flows that satisfy the constraints for all F of size at most $r/((1+\epsilon)k)$.

THEOREM 2.2. *For every $\epsilon > 0$ there is a polynomial time algorithm that, given an instance of the unit edge-length directed r -fault-tolerant k -spanner problem, finds a set of fractional capacities $\{x_e\}_{e \in E}$ with the following two properties: 1) there exist flow variables that satisfy the flow and capacity constraints of LP (4) for all fault sets of size at most $\frac{r}{(1+\epsilon)k}$, and 2) $\sum_{e \in E} x_e$ is at most $\frac{1+\epsilon}{\epsilon}$ times larger than the optimal solution to LP (4).*

PROOF. When r is super-constant the number of constraints in the LP is super-polynomial, so we cannot solve it using earlier methods (when we transform to the dual we get a super-polynomial number of variables). Instead of going through the dual we will stick with the primal and give a separation oracle. However, since ellipsoid with a separation oracle takes time polynomial in the dimension (i.e. the number of variables) we need to transform the problem into one with a polynomial number of variables. We do this in a simple way: we simply project the polytope down on the capacity variables x_e , of which there are only $O(m)$. The objective function of LP (4) uses only the x_e variables, so optimizing over this projection is sufficient to optimize over the full LP. And since this is a projection of a convex set it is itself convex, so if we can design a separation oracle the ellipsoid algorithm will run in polynomial time.

So what would a separation oracle for this projected polytope be? Simply examining LP (4) shows that a setting of the capacity variables $\{x_e\}_{e \in E}$ is not a valid solution if and only if there is some set of at most r faults such that it is impossible to send 1 unit of flow between all demands. Slightly more formally, $\{x_e\}_{e \in E}$ is not a valid solution if and only if there is some fault set F (of size at most r) and edge $(u, v) \in E$ such that:

1. $(u, v) \notin F$ for the edge-fault case, or $u, v \notin F$ for the vertex-fault case, and
2. There is no way of sending one unit of flow along stretch- k paths from u to v in $G \setminus F$ while respecting capacities $\{x_e\}$.

By strong duality, the maximum flow that can be sent along $u - v$ paths of stretch at most k is equal to the smallest fractional cut, where a fractional cut is an assignment of values y_e to the edges such that $\sum_{e \in P} y_e \geq 1$ for all $P \in \mathcal{P}_{u,v}^F$. The size of such a cut for a particular fault set F is $\sum_{e \in E} x_e y_e$. So for every fault set F of size at most r , for every remaining demand (u, v) , for every fractional cut $\{y_e\}_{e \in E}$ relative to F and to (u, v) , any feasible solution $\{x_e\}_{e \in E}$ has $\sum_{e \in E} y_e x_e \geq 1$. These are the violated constraints that our separation oracle will find (or at least will approximately find).

So to construct a separation oracle, we want to find a set of faults F and demand (u, v) with the smallest fractional

cut. If the size of this cut is less than 1, then we have found a separating hyperplane, and if there is no such set F and demand (u, v) then the current capacities are feasible. In order to solve this problem, which we will call **STRETCH- k INTERDICTION**, we first write it as a mixed-integer program. Since there are only a polynomial number of (u, v) demands we can simply try them all, so our formulation is for some given (u, v) . This formulation is for the vertex-fault version; the edge-fault version follows the same basic idea. Recall that the x_e 's are the capacity variables in the original problem, so in this context they are fixed constants and thus the objective function and the constraints are linear. The intention of MIP (5) is for z_w to represent whether vertex w is part of the fault set and for the $\{y_e\}$ variables to represent a fractional cut of the remaining paths.

$$\begin{array}{ll}
 \min & \sum_{e \in E} x_e y_e \\
 \text{s.t.} & \sum_{(a,b) \in P} (y_{(a,b)} + \frac{1}{2} z_a + \frac{1}{2} z_b) \geq 1 \quad \forall P \in \mathcal{P}_{u,v} \\
 & \sum_{w \in V} z_w \leq r \\
 & z_u = z_v = 0 \\
 & z_w \in \{0, 1\} \quad \forall w \in V \\
 & y_e \geq 0 \quad \forall e \in E
 \end{array} \tag{5}$$

CLAIM 2.3. *MIP (5) is an exact formulation of **STRETCH- k INTERDICTION**.*

PROOF. Note that there is a one-to-one correspondence between settings of the z_w variables and possible fault sets. For every setting of the z_w 's, the objective value is the minimum cost fractional cut (where we have to cut stretch- k paths that do not hit any faults), which is exactly what we are trying to optimize. \square

LEMMA 2.4. *There is a bicriteria approximation for the **STRETCH- k INTERDICTION** problem that uses at most $(1 + \epsilon)kr$ faults (instead of r) and has cost at most $\frac{1+\epsilon}{\epsilon}$ times the best r -fault solution.*

PROOF. In order to solve MIP (5) we relax the integrality constraints on the z_w variables to $0 \leq z_w \leq 1$, giving us a linear program. We can solve the resulting LP by constructing its own separation oracle: if we define the length of an edge (a, b) to be $y_{(a,b)} + \frac{1}{2} z_a + \frac{1}{2} z_b$, is there a stretch- k path with length less than 1? Since we only consider the fault-tolerant setting for the unit-length case we can actually solve this problem exactly using Bellman-Ford (since in this case stretch- k is equivalent to k -hop). So we can solve this LP in polynomial time.

Now we need to round the z_w variables to integers. We will use a very simple threshold rounding: if $z_w \geq \frac{1}{(1+\epsilon)k}$ then set $z'_w = 1$; otherwise set $z'_w = 0$. Furthermore, set $y'_e = \frac{1+\epsilon}{\epsilon} y_e$. Since any stretch- k path has at most k hops, if $\sum_{w \in P} z_w \geq \frac{1}{1+\epsilon}$ then $z_w \geq \frac{1}{(1+\epsilon)k}$ for some $w \in V$, and thus z'_w covers P . On the other hand, if $\sum_{w \in P} z_w < \frac{1}{1+\epsilon}$ then $\sum_{e \in P} y_e > \frac{\epsilon}{1+\epsilon}$, so $\sum_{e \in P} y'_e = \frac{1+\epsilon}{\epsilon} \sum_{e \in P} y_e \geq 1$ and the y' variables cover P . Thus (z', y') is a valid solution to the MIP except that $\sum_{w \in V} z'_w \leq (1 + \epsilon)kr$ instead of being at

most r . In other words, we have designed a $((1 + \epsilon)k, \frac{1+\epsilon}{\epsilon})$ -bicriteria approximation algorithm for MIP (5) and thus for STRETCH- k INTERDICTION. \square

By using this bicriteria approximation with original fault budget $r/(1 + \epsilon)k$ instead of r , we will find a separating hyperplane (whose coefficients are the $\{y'_e\}$ variables) as long as there is some fault set F of size at most $r/(1 + \epsilon)k$ and demand (u, v) for which the maximum stretch- k flow (or equivalently the minimum fractional cut) is at most $\epsilon/(1 + \epsilon)$. So using this separation oracle with the Ellipsoid algorithm and then rounding the capacities we find up by $\frac{1+\epsilon}{\epsilon}$ gives us a bicriteria algorithm for LP (4), yielding the theorem. \square

3. APPROXIMATIONS FOR DIRECTED k -SPANNER

We will now design a $\tilde{O}(n^{2/3})$ -approximation algorithm for the directed k -spanner problem. We first solve LP (2) as detailed in Theorem 2.1 to get a fractional solution (x, f) . We then round this solution using Algorithm 1, which has two main components: a simple threshold rounding scheme together with a collection of shortest path arborescences. This technique of combining LP rounding with shortest-path arborescences has been used before, notably by Bhat-tacharyya et al. [BGJ⁺09], so our main contribution is the new LP and rounding scheme.

Algorithm 1: Rounding Algorithm for Directed k -Spanner

```

1  $E' \leftarrow \{e \in E : x_e \geq 1/(3n \ln n)^{2/3}\}$ 
2 for  $i \leftarrow 1$  to  $(3n \ln n)^{2/3}$  do
3   Choose  $v \in V$  uniformly at random
4    $T_i^{in} \leftarrow$  shortest path in-arborescence rooted at  $v$ 
5    $T_i^{out} \leftarrow$  shortest path out-arborescence rooted at  $v$ 
6 Output  $E' \cup \left( \bigcup_{i=1}^{(3n \ln n)^{2/3}} (T_i^{in} \cup T_i^{out}) \right)$ 
```

To show that this algorithm gives a valid k -spanner, we begin with a lemma that characterizes edges that are satisfied by the thresholding. For every (u, v) in E , let $N_{u,v} \subseteq V$ be the set of vertices that lie on a path of stretch at most k from u to v (i.e. the set of vertices that are used by at least one path in $\mathcal{P}_{u,v}$).

LEMMA 3.1. *For every $(u, v) \in E$ there is a path $P \in \mathcal{P}_{u,v}$ with the property that every edge $e \in P$ has $x_e \geq \frac{1}{|N_{u,v}|^2}$.*

PROOF. Suppose this is false for some (u, v) . Let $B \subseteq N_{u,v} \times N_{u,v}$ be the set of edges with $x_e < 1/|N_{u,v}|^2$. Then every path $P \in \mathcal{P}_{u,v}$ goes through at least one edge in B , so these edges form a cut between u and v relative to the paths in $\mathcal{P}_{u,v}$. Since we have a valid LP solution, we know that at least one unit of flow is sent from u to v using paths in $\mathcal{P}_{u,v}$. This means that the number of edges in B must be at least $|N_{u,v}|^2$. But this is a contradiction: every edge in B has both endpoints in $N_{u,v}$, so there are at most $\binom{|N_{u,v}|}{2} < |N_{u,v}|^2$ of them. \square

So if $|N_{u,v}|$ is small, Lemma 3.1 implies that there is some stretch k path with the property that every edge is assigned a large capacity. On the other hand, if $|N_{u,v}|$ is large then

there are many nodes that are on stretch k paths, so we should be able to find such a path by picking nodes randomly. This is formalized in the following lemma:

LEMMA 3.2. *If we sample at least $\frac{3n \ln n}{|N_{u,v}|}$ vertices independently and uniformly at random, then with probability at least $1 - 1/n^3$ at least one sampled vertex will be in $N_{u,v}$.*

PROOF. The probability that no sampled vertex is in $N_{u,v}$ is at most $\left(1 - \frac{|N_{u,v}|}{n}\right)^{\frac{3n \ln n}{|N_{u,v}|}} \leq e^{-3 \ln n} = 1/n^3$ and thus the probability that at least one sampled vertex is in $N_{u,v}$ is at least $1 - 1/n^3$. \square

THEOREM 3.3. *There is a polynomial time algorithm that is given a directed graph G with nonnegative edge lengths and $k \geq 1$ and with high probability returns a directed k -spanner of size at most $O((n \ln n)^{2/3})$ times the smallest directed k -spanner.*

PROOF. The algorithm is simply to solve LP (2) using Theorem 2.1, and then round the solution using Algorithm 1. We first prove that it results in a valid spanner with high probability. Consider some edge $(u, v) \in E$. If $|N_{u,v}| \leq (3n \ln n)^{1/3}$, then Lemma 3.1 implies that there is some stretch k path from u to v using edges contained in E' and thus in the spanner. On the other hand, if $|N_{u,v}| > (3n \ln n)^{1/3}$ then Lemma 3.2 implies that with probability at least $1 - 1/n^3$ we will have sampled some vertex in $N_{u,v}$. Suppose we sample $w \in N_{u,v}$ on the i th iteration. By the definition of $N_{u,v}$ we know that w is on some path from u to v with stretch at most k , and thus the length of the shortest path from u to w plus the length of the shortest path from w to v is at most $k \cdot d_G(u, v)$. These paths are contained in $T_i^{in} \cup T_i^{out}$, so the spanner will include both of these shortest paths and thus will include a path from u to v with stretch at most k . Taking a union bound over all (u, v) completes the proof that the returned subgraph is a k -spanner.

To prove that it is a $O((n \ln n)^{2/3})$ -approximation we will show that each of the two steps costs at most $O((n \ln n)^{2/3}) \times OPT$. This is obvious for the LP rounding step: an edge e is in E' only if $x_e \geq 1/(3n \ln n)^{2/3}$, so $|E'|$ is at most $O((n \ln n)^{2/3})$ times the LP cost. To show that the second step does not add many edges, note that every iteration adds at most $2(n - 1)$ edges, and that $n - 1$ is a trivial lower bound on OPT (since we are assuming the underlying graph is connected; if it is not connected then it is easy to modify this analysis to still hold). Thus the total cost of all the arborescences is at most $2(3n \ln n)^{2/3} \times OPT$. \square

3.1 Extension to Fault-Tolerant Version

In order to adapt the rounding scheme of Algorithm 1 to the fault-tolerant case we need to show how to modify the threshold rounding and the arborescence sampling. It is simple to see that Lemma 3.1 still holds for every fault set, so the threshold rounding will still work (although we will change the threshold). But the arborescence rounding must be changed to allow for faults. The intuition behind the change comes from the technique of color-coding [AYZ95]: before randomly sampling the root of an arborescence, independently fail each element (either edges in the edge-tolerant version or vertices in the vertex-tolerant version) with some probability p . We call this technique *failure sampling*. We then randomly sample a root and include its

shortest-path in- and out-arborescences in the resulting subgraph. By setting p , the number of arborescences sampled, and the threshold of the rounding appropriately, we get the following theorem. We say that an algorithm is an (α, β) -approximation for the r -fault-tolerant directed k -spanner problem if it returns a αr -fault-tolerant k -spanner of size at most β times the smallest k -spanner, and an algorithm is a *true* β -approximation if it is a $(1, \beta)$ -approximation.

THEOREM 3.4. *For every constant $\epsilon > 0$ there is a polynomial time algorithm that is a*

$$\left(\frac{1}{(1+\epsilon)k}, O\left(\left(\frac{(1+\epsilon)r(k+r)^{k+r}n \ln n}{\epsilon r^r k^k}\right)^{2/3}\right) \right)$$

approximation for r -fault-tolerant directed k -spanner with unit edge-lengths. There is also a true approximation algorithm with the same approximation ratio that takes $n^{O(r)}$ time.

PROOF. Suppose that we have a feasible solution for the fault-tolerant relaxation LP (4) (or at least an approximate solution from Theorem 2.2). We show how to round a fractional solution to an integer solution, assuming unit edge lengths. Our rounding is basically the same as for the non-fault tolerant version. In particular, Lemma 3.1 still holds for every fault set. So we can, as before, set a threshold value t and round up any edge with $x_e \geq 1/t$. The only difference comes in the random sampling step: in the non-fault-tolerant version, it sufficed to randomly pick centers of shortest path in- and out-arborescences. But in the fault-tolerant setting that is no longer sufficient; the paths we construct must suffice even after failures, which simple shortest paths obviously will not. So we will add an extra step inspired by *color-coding* [AYZ95]: before randomly sampling the root of an arborescence, independently fail each element (either edges in the edge-tolerant version or vertices in the vertex-tolerant version) with some probability p . We call this technique *failure sampling*. We then randomly sample the root of shortest-path in- and out-arborescences in the resulting subgraph.

Consider some fault set F and some edge (u, v) that still survives in $G \setminus F$. Define $N_F(u, v)$ as in the no-fault setting: a vertex x is in $N_F(u, v)$ if x is on some stretch- k $u-v$ path in $G \setminus F$. Since Lemma 3.1 still holds, if $|N_F(u, v)| \leq \sqrt{t}$ then the threshold rounding satisfies the demand. So we assume that $|N_F(u, v)| > \sqrt{t}$ and analyze the probability that a single round of the random sampling will satisfy the demand. We will then perform the number of rounds necessary to be able to take a union bound over all possible F and (u, v) .

Since we are assuming all edge lengths are 1, a stretch- k path is the same as a path with at most k hops. A sufficient condition for the sampling to succeed for F and (u, v) is for the arborescence root to be a vertex in $N_F(u, v)$, everything in F to be killed by the failure sampling, and nothing from the k -hop path containing the root to be killed (note that such a path must exist by the definition of $N_F(u, v)$). Given that the root is selected to be in $N_F(u, v)$, the probability that the particular k -hop $u-v$ path containing the root is all preserved by the failure sampling is $(1-p)^k$. And clearly the probability that everything from F is killed by the failure sampling is p^r , and is independent of the other two events. So the probability that all three events happen is $p^r \times \frac{|N_F(u, v)|}{n} \times (1-p)^k \geq p^r (1-p)^k \frac{\sqrt{t}}{n}$. The number of

possible failure sets F and demands (u, v) is at most $\binom{m}{r} \times \binom{n}{2} \leq n^{2r+2}$ in the edge-failure setting; for vertex failures it is at most n^{r+2} . Let ℓ be the number of rounds for which we repeat the random sampling. Then in order to succeed on all constraints with probability at least $1/2$, we want

$$\left(1 - \frac{p^r (1-p)^k \sqrt{t}}{n}\right)^\ell \leq \frac{1}{2n^{2r+2}}$$

Setting $p = \frac{r}{k+r}$ and solving for ℓ , we get that it is sufficient to set

$$\ell = \frac{2(2r+2)(k+r)^{k+r}n \ln n}{r^r k^k \sqrt{t}}$$

As in the non-fault-tolerant case, we balance out the cost of the sampling (ℓ) with the cost of the threshold rounding (t) to get a total approximation of

$$O\left(\left(\frac{r(k+r)^{k+r}n \ln n}{r^r k^k}\right)^{2/3}\right)$$

If r is constant, then this rounding combined with our ability to actually solve LP (4) gives us the theorem for the $r = O(1)$ case. If r is not constant then we need to use Theorem 2.2 before the rounding procedure, giving us the claimed bicriteria approximation. \square

3.2 Integrality Gap

We now want to complement our approximation algorithm by proving a nearly matching integrality gap. We do this by a reduction from the MIN-REP problem. In MIN-REP we are given a bipartite graph $G = (U, V, E)$ together with a partition of U and V into groups U_1, U_2, \dots, U_p and V_1, V_2, \dots, V_p . We say that there is a *superedge* between two groups (U_i, V_j) if there is some $u \in U_i$ and some $v \in V_j$ such that $\{u, v\} \in E$. The goal is to find a subset $X \subseteq U \cup V$ of as few vertices as possible such that for every pair of groups (U_i, V_j) with a superedge there is some $u \in U_i \cap X$ and $v \in V_j \cap X$ such that $\{u, v\} \in E$.

Elkin and Peleg [EP07] proved hardness of approximation for directed k -spanner by using a reduction from MIN-REP, and we will use their reduction to prove an integrality gap. But instead of reducing from generic MIN-REP instances as in a hardness proof, we will only apply the reduction to instances of MIN-REP in which every superedge actually corresponds to a matching between vertices, i.e. if (U_i, V_j) is a superedge then there is a matching between U_i and V_j . The interested reader might note that these are basically instances of the *Unique Games Problem* [Kho02].

We first give a lemma that is implicit in the integrality gap for Label Cover recently proved by Charikar, Hajiaghayi, and Karloff [CHK09]:

LEMMA 3.5. *For every constant $\epsilon > 0$, there are instances of MIN-REP with the following properties:*

1. *Every group has size $n^{\frac{2}{3}-\epsilon}$,*
2. *$OPT \geq n^{\frac{2}{3}-\epsilon}$,*
3. *There is a matching between every U_i and V_j .*

We now use this lemma to prove the main integrality gap theorem by applying the Elkin and Peleg reduction [EP07] to instances from Lemma 3.5 and showing there it has a small fractional solution.

THEOREM 3.6. *The integrality gap of LP (2) on the unit-length directed k -spanner problem is $\Omega(\frac{1}{k}n^{\frac{1}{3}-\epsilon})$ for every constant $\epsilon > 0$.*

PROOF. The instances that we use to prove this integrality gap are the instances we obtain by applying the reduction of Elkin and Peleg [EP07] to the instances from Lemma 3.5. We explain the reduction in detail so as to analyze the best fractional LP solution. Let r be the number of groups, so in the instances from Lemma 3.5 we have that $r = n^{\frac{1}{3}+\epsilon}$. For each group we will add $x = n^{\frac{2}{3}-\epsilon}/((k-1)/2)$ paths, where each path has length $(k-1)/2$ (for ease of exposition we assume that k is odd, but it does not actually matter). More formally, let (U', V', E') be a MIN-REP instance from Lemma 3.5, and let $n_{MR} = |U' \cup V'|$. Then the vertex set of our spanner problem is

$$V = (U' \cup V') \cup \left(\bigcup_{p=1}^x \bigcup_{i=1}^{r/2} \bigcup_{j=1}^{(k-1)/2} s_{i,j}^p \right) \cup \left(\bigcup_{p=1}^x \bigcup_{i=1}^{r/2} \bigcup_{j=1}^{(k-1)/2} t_{i,j}^p \right).$$

Note that $n = |V| = n_{MR} + x \cdot r \cdot (k-1)/2 = n_{MR} + ((n^{\frac{2}{3}-\epsilon}/((k-1)/2)) \cdot n^{\frac{1}{3}+\epsilon} \cdot ((k-1)/2)) = 2n_{MR}$, so we have only doubled the number of vertices.

The edge set is divided into a few different components. First, let $E'' = \{(u, v) : u \in U', v \in V', \{u, v\} \in E'\}$ be the original MIN-REP edges but now directed from U to V . Next we add a clique to every group: let $E_C = \bigcup_{i=1}^{r/2} ((U_i \times U_i) \cup (V_i \times V_i))$. We also want to turn the new vertices into paths: let

$$E_M = \bigcup_{p=1}^x \bigcup_{i=1}^{r/2} \bigcup_{j=1}^{(k-3)/2} \{(s_{i,j}^p, s_{i,j+1}^p), (t_{i,j}^p, t_{i,j+1}^p)\}$$

so for every fixed i and p , the E_M edges form a directed path from $s_{i,1}^p$ to $s_{i,(k-1)/2}^p$, and similarly for the t vertices. We also add edges to connect these paths to the original vertices: let E_U equal

$$\bigcup_{p=1}^x \bigcup_{i=1}^{r/2} \left(\{(s_{i,(k-1)/2}^p, u) : u \in U_i\} \cup \{(v, t_{i,1}^p) : v \in V_i\} \right).$$

Finally, we want to add edges to connect the endpoints of paths corresponding to superedges (which in our case is all (U_i, V_j)): let

$$E_I = \bigcup_{p=1}^x \bigcup_{i=1}^{r/2} \bigcup_{j=1}^{r/2} \{(s_{i,1}^p, t_{j,(k-1)/2}^p)\}.$$

So our final edge set is $E = E'' \cup E_C \cup E_M \cup E_U \cup E_I$.

Elkin and Peleg [EP07] showed that the optimal spanner has size at least $\Omega(x \times OPT_{MR})$, where OPT_{MR} is the size of the smallest MIN-REP solution. So in our case, the best spanner has size at least

$$\Omega \left(\frac{n^{\frac{2}{3}-\epsilon}}{(k-1)/2} \times n^{\frac{2}{3}-\epsilon} \right) = \Omega \left(\frac{1}{k} n^{\frac{4}{3}-2\epsilon} \right).$$

On the other hand, we claim that the best fractional solution is small, namely $O(n)$. To see this, consider the following fractional assignment. All edges inside the tails, i.e. all edges in E_M , have fractional capacity 1. Let $q = n_{MR}/r =$

$n^{\frac{2}{3}-\epsilon}$ be the size of each group. We set the fractional capacity of all edges in $E'' \cup E_C \cup E_U$ to $2/q$, and set the fractional capacity of all edge in E_I to 0. The cost of this solution is

$$\begin{aligned} & |E_M| + \frac{2}{q} (|E''| + |E_C| + |E_U|) \\ & \leq xr \frac{k-3}{2} + \frac{2}{q} \left(\frac{r^2 q}{4} + q^2 r + xrq \right) \\ & \leq xr \frac{k-3}{2} + \frac{2r^2}{4} + 2qr + 2xr \\ & \leq n_{MR} + n_{MR}^{\frac{1}{3}+\epsilon} + 2n_{MR} + 2n_{MR} \\ & = O(n). \end{aligned}$$

So it remains to prove that it is a valid fractional solution. We proceed by analyzing each type of edge. Obvious since every edge in E_M is included with capacity 1, we are able to send one unit of flow. For some edge $(u, v) \in E''$, we can send $q-1$ flows, each of size $1/q$, first to the vertices in the same group as u (via the E_C edges), then across the matching to the group containing v (via E'' edges), and then back to v (via E_C edges). We can send the final $1/q$ flow directly on the edge (u, v) . These paths have length $3 \leq k$ and obviously satisfy capacity constraints. For some edge $(a, b) \in E_C$, we can do the same thing without crossing any matching: send $1/(q-1) \leq 2/q$ flow to each of the group mates of a , and then back into b . For an edge $(s_{i,(k-1)/2}^p, u) \in E_U$, we can send $1/q$ flow from $s_{i,(k-1)/2}^p$ to each of the vertices in the group containing u (using E_U edges) and the from those vertices to u (using E_C edges). Similarly, for an edge $(v, t_{i,1}^p)$ we can send $1/(q-1) < 2/q$ flow to each of the other vertices in V_i , and then from these vertices to $t_{i,q}^p$. Finally, for an edge $(s_{i,1}^p, t_{j,(k-1)/2}^p) \in E_I$ we can send one unit along the $s_{i,\dots}^p$ path, then split it into q paths using the E_U edges, send each of those $1/q$ flows across the matching using E'' edges, recombine the flow at $t_{j,1}^p$ using E_U edges, and then send it down the path to $t_{j,(k-1)/2}^p$. Each of these q paths has length k and satisfies the fractional capacities. Thus this is a valid fractional solution. \square

The reason that we lose an extra $\frac{1}{k}$ in the integrality gap is that because we are in the unit-length case we need to add many vertices in order to build long paths. It is easy to modify the proof of Theorem 3.6 to give a gap of $\Omega(n^{\frac{1}{3}-\epsilon})$ in the arbitrary lengths setting.

4. DIRECTED UNIT-LENGTH 3-SPANNER

While our $\tilde{O}(n^{2/3})$ -approximation is an improvement over previous work for arbitrary edge lengths and for unit edge lengths with $k > 3$, for unit edge lengths with $k = 3$ it matches the previous bounds of Elkin and Peleg [EP05] and Bhattacharyya et al. [BGJ⁺09]. So for the specific case of unit-length directed 3-spanner we develop a different rounding algorithm for the flow-based LP (2) that gives an $\tilde{O}(\sqrt{n})$ -approximation. Our algorithm first solves LP (2) and then rounds it using Algorithm 2. Informally, this rounding works by choosing a threshold value $T_v \in [0, 1]$ for each vertex $v \in V$. We then add all edges (u, v) where either T_u or T_v is at most $\rho x_{u,v}$, where $\rho = \Theta(\sqrt{n \log n})$ is an inflation factor to make the probabilities large enough. This turns out to not be quite enough edges, so we also add all edges (u, v) where both T_u and T_v are at most $\sqrt{\rho x_{u,v}}$. For technical reasons we have to add an extra complication: every vertex will

actually choose another threshold, T'_v , and edges are added as described for every combination of T and T' thresholds.

Algorithm 2: Rounding Algorithm for 3-spanner

- 1 Set $\rho = C\sqrt{n} \log n$ for a large constant C
 - 2 For every $v \in V$ choose independently two values $T_v, T'_v \in_R [0, 1]$
 - 3 Let $E_1 = \{(u, v) \in E : \min\{T_u, T'_u, T_v, T'_v\} \leq \rho \cdot x_{u,v}\}$
 - 4 Let $E_2 = \{(u, v) \in E : \max\{\min\{T_u, T'_u\}, \min\{T_v, T'_v\}\} \leq \sqrt{\rho \cdot x_{u,v}}\}$
 - 5 Output $E' = E_1 \cup E_2$
-

LEMMA 4.1. *Algorithm 2 returns a set of edges E' with $\mathbb{E}[|E'|] \leq O(\rho)$ times the size of the smallest 3-spanner.*

PROOF. See [DK10]. \square

LEMMA 4.2. *For every edge $(u, v) \in E$, the probability that there is no path of length at most 3 from u to v in E' is at most $1/e$.*

PROOF. We provide only a sketch; see [DK10] for details. If a significant amount of flow is sent along paths of length 1 or 2, then the analysis is trivial. If most of the flow is sent along paths of length 3, then we break the paths into classes depending on whether it is better to use E_1 or E_2 when analyzing each hop and then prove that at whichever class has the most flow contributes at least one path with probability at least $1/e$. In some cases we analyze the expected number of paths of length 3 and then control correlations using concentration bounds such as Janson’s inequality [DP09, Chapter 3]. But in other cases this type of analysis is not effective, so we “decompose” the paths into their first, second and third hops and then use standard concentration bounds separately for each hop plus some global arguments based on flow conservation. \square

It is interesting to note that Lemmas 4.1 and 4.2 hold even for a weighted version in which every edge has an arbitrary nonnegative cost associated with it and our goal is to find the minimum cost 3-spanner. So our approximation algorithm actually generalizes to this weighted version.

THEOREM 4.3. *There is a polynomial time $\tilde{O}(\sqrt{n})$ approximation algorithm for the unit-length directed 3-spanner problem, even with arbitrary costs on the edges.*

PROOF. The algorithm is simple: solve LP (2) (note that this can be done exactly since in this setting the linear program has only a polynomial number of variable) and then repeat Algorithm 2 with fresh randomness $O(\log n)$ times. Lemma 4.2 implies that this gives a valid solution with high probability, and Lemma 4.1 implies that it is a $\tilde{O}(\sqrt{n})$ -approximation. \square

4.1 Extension to Fault-Tolerant Version

It is easy to see that this algorithm can be trivially extended to the r -fault-tolerant setting. For each set F of faults, the analysis works the same as in Lemma 4.2. We just need to solve LP (4) instead of LP (2) and change the parameters (ρ and the number of times Algorithm 2 is repeated) to make the probability of failure small enough to apply a union bound to all possible failure sets F and

edges (u, v) instead of just over the edges. In particular, for the vertex failure setting we need the probability of failure to be less than $1/n^{r+2}$ and for the edge failure setting we need to probability of failure to be less than $1/n^{2r+2}$. The main takeaway is that for directed 3-spanner we get fault-tolerance at a cost of only $\tilde{O}(r)$ instead of something exponential in r , as for the $k > 3$ case and previous work on absolute bounds [CLPR09].

THEOREM 4.4. *For every constant $\epsilon > 0$ there is a polynomial time $\left(\frac{1}{3(1+\epsilon)}, O\left(\frac{1+\epsilon}{\epsilon} r \sqrt{n} \log^2 n\right)\right)$ -approximation algorithm for the r -fault-tolerant directed 3-spanner problem with unit lengths. There is also a true approximation algorithm with approximation ratio $O(r\sqrt{n} \log^2 n)$ that takes $n^{O(r)}$ time.*

PROOF. See [DK10]. \square

5. UNIT-LENGTH 2-SPANNER

The 2-spanner problem is qualitatively and quantitatively different from k -spanner with $k > 2$: it is known [KP94, EP01] that it can be approximated to $O(\log n)$ and that this is tight [Kor01] (assuming $P \neq NP$). Note that this approximation algorithm is only known to work for the unit-length version, but does allow the edges to have arbitrary costs with the goal of finding a 2-spanner of minimum cost. We show that our LP relaxation (2) has integrality gap of $\Theta(\log n)$, and thus offers comparable approximation ratio. We further show that our approach, namely the flow-based LP relaxation and a direct rounding procedure, easily adapts to bounded-degree case which was studied in the literature [KP94, DK99, EP01], and also to the fault-tolerant case, for which no approximation was previously known. All proofs can be found in the full version [DK10].

To prove a lower bound on the integrality gap, we will apply the hardness reduction from set cover to 2-spanner to an instance of set cover that has a large integrality gap. This gives the following theorem:

THEOREM 5.1. *The integrality gap of LP (2) for the 2-spanner problem with unit edge-lengths is $\Omega(\log n)$ even when all edge costs are unit.*

We give an upper bound on the integrality gap of the LP by randomized rounding of the LP solution. A simplified version of the rounding scheme for 3-spanner described in Algorithm 2 suffices to give an $O(\log n)$ -approximation, even for the case when the edges have costs. If the edges do not have costs, then by using constructive versions of the Lovász Local Lemma we can improve this to $O(\log \Delta)$ (where Δ is the maximum degree in G). This gives the following theorem:

THEOREM 5.2. *For the directed 2-spanner problem with unit edge-lengths, LP (2) has integrality gap $O(\log n)$ with edge costs and $O(\log \Delta)$ if all edge costs are unit.*

It is easy to extend these results to the r -fault-tolerant version of the problem, with respect to the LP relaxation (4) (which in this setting we can actually solve). We will have to pay only an extra multiplicative r in the approximation.

THEOREM 5.3. *For the r fault-tolerant version of the directed 2-spanner problem with unit edge-lengths, the LP relaxation (4) has integrality gap $O(r \log n)$ with edge costs and $O(r \log \Delta)$ if all edge costs are unit.*

6. REFERENCES

- [ADD⁺93] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9(1):81–100, 1993.
- [AP95] B. Awerbuch and D. Peleg. Online tracking of mobile users. *J. ACM*, 42(5):1021–1058, 1995.
- [AS00] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience [John Wiley & Sons], New York, second edition, 2000.
- [AYZ95] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.
- [BGJ⁺09] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff. Transitive-closure spanners. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 932–941, 2009.
- [BRR10] P. Berman, S. Raskhodnikova, and G. Ruan. Finding Sparser Directed Spanners. In *FSTTCS 2010*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 424–435, 2010.
- [CHK09] M. Charikar, M. Hajiaghayi, and H. J. Karloff. Improved approximation algorithms for label cover problems. In *European Symposium on Algorithms*, pages 23–34, 2009.
- [CLPR09] S. Chechik, M. Langberg, D. Peleg, and L. Roditty. Fault-tolerant spanners for general graphs. In *41st Annual ACM Symposium on Theory of Computing*, pages 435–444. ACM, 2009.
- [DK99] Y. Dodis and S. Khanna. Design networks with bounded pairwise distance. In *31st Annual ACM Symposium on Theory of Computing*, pages 750–759. ACM, 1999.
- [DK10] M. Dinitz and R. Krauthgamer. Directed spanners via flow-based linear programs. *CoRR*, abs/1011.3701, 2010.
- [DK11] M. Dinitz and R. Krauthgamer. Fault-tolerant spanners: Better and simpler. In *PODC*, 2011.
- [DP09] D. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 2009.
- [EEST08] M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng. Lower-stretch spanning trees. *SIAM J. Comput.*, 38(2):608–628, 2008.
- [EP01] M. Elkin and D. Peleg. The client-server 2-spanner problem with applications to network design. In *8th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 117–132, 2001.
- [EP05] M. Elkin and D. Peleg. Approximating k -spanner problems for $k > 2$. *Theor. Comput. Sci.*, 337(1-3):249–277, 2005.
- [EP07] M. Elkin and D. Peleg. The hardness of approximating spanner problems. *Theor. Comp. Sys.*, 41(4):691–729, 2007.
- [Has92] R. Hassin. Approximation schemes for the restricted shortest path problem. *Math. Oper. Res.*, 17(1):36–42, 1992.
- [Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *34th Annual ACM Symposium on the Theory of Computing*, pages 767–775, July 2002.
- [Kor01] G. Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.
- [KP94] G. Kortsarz and D. Peleg. Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994.
- [LR01] D. H. Lorenz and D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213 – 219, 2001.
- [PS89] D. Peleg and A. A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.
- [RT87] P. Raghavan and C. D. Thompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.
- [ST04] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *36th Annual ACM Symposium on Theory of Computing*, pages 81–90. ACM, 2004.
- [TZ05] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.