Transitive-Closure Spanners*

Arnab Bhattacharyya† Elena Grigorescu† Kyomin Jung† Sofya Raskhodnikova‡
David P. Woodruff§

Abstract

We define the notion of a transitive-closure spanner of a directed graph. Given a directed graph $G = (V, E)$ and an integer $k \geq 1$, a *k-transitive-closure-spanner (k-TC-spanner)* of G is a directed graph $H = (V, E_H)$ that has (1) the same transitive-closure as G and (2) diameter at most k. These spanners were studied implicitly in access control, property testing, and data structures, and properties of these spanners have been rediscovered over the span of 20 years. We bring these areas under the unifying framework of TC-spanners. We abstract the common task implicitly tackled in these diverse applications as the problem of constructing sparse TC-spanners.

We study the approximability of the size of the sparsest k-TC-spanner for a given digraph. Our technical contributions fall into three categories: algorithms for general digraphs, inapproximability results, and structural bounds for a specific graph family which imply an efficient algorithm with a good approximation ratio for that family.

Algorithms. We present two efficient deterministic algorithms that find k-TC-spanners of near optimal size. The first algorithm gives an $\tilde{O}(n^{1-1/k})$-approximation for $k > 2$. Our method, based on a combination of convex programming and sampling, yields the first sublinear approximation ratios for (1) DIRECTED k-SPANNER, a well-studied generalization of k-TC-SPANNER, and (2) its variants CLIENT/SERVER DIRECTED k-SPANNER, and the k-DIAMETER SPANNING SUBGRAPH. This resolves the main open question of Elkin and Peleg (IPCO, 2001). The second algorithm, specific to the k-TC-spanner problem, gives an $\tilde{O}(n/k^2)$-approximation. It shows that for $k = \Omega(\sqrt{n})$, our problem has a provably better approximation ratio than DIRECTED k-SPANNER and its variants. This algorithm also resolves an open question of Hesse (SODA, 2003).

Inapproximability. Our main technical contribution is a pair of strong inapproximability results. We resolve the approximability of 2-TC-spanners, showing that it is $\Theta(\log n)$ unless $P = NP$. For constant $k \geq 3$, we prove that the size of the sparsest k-TC-spanner is hard to approximate within $2^{\log^{1-\epsilon} n}$, for any $\epsilon > 0$, unless NP $\subseteq$ DTIME$(n^{\text{polylog}\, n})$. Our hardness result helps explain the difficulty in designing general efficient solutions for the applications above, and it cannot be improved without resolving a long-standing open question in complexity theory. It uses an involved application of generalized butterfly and broom graphs, as well as noise-resilient transformations of hard problems, which may be of independent interest.

Structural bounds. Finally, we study the size of the sparsest TC-spanner for H-minor-free digraphs, which include planar, bounded genus, and bounded tree-width graphs, explicitly investigated in applications above. We show that every H-minor-free digraph has an efficiently constructible k-TC-spanner of size $\tilde{O}(n)$. This implies an $\tilde{O}(1)$-approximation algorithm for this family. Furthermore, using our insight that 2-TC-spanners yield property testers, we obtain a monotonicity tester with $O(\log^2 n/\epsilon)$ queries for any poset whose transitive reduction is an H-minor free digraph. This improves and generalizes the previous $\Theta(\sqrt{n}\log n/\epsilon)$-query tester of Fischer *et al* (STOC, 2002).

1 Introduction

A *spanner* can be thought of as a sparse backbone of a graph that approximately preserves distances between every pair of vertices. More precisely, a subgraph $H = (V, E_H)$ is a *k-spanner* of $G = (V, E)$ if for every pair of vertices $u, v \in V$, the shortest path distance $d_H(u, v)$ from u to v in H is at most $k \cdot d_G(u, v)$. Since they were introduced by Peleg and Schäffer [36] in the context of distributed computing, spanners for undirected graphs have been extensively studied. The tradeoff between the parameter k, called the *stretch*, and the number of edges in a spanner is relatively well understood: for every $k \geq 1$, any undirected graph on n vertices has a $(2k-1)$-spanner with $O(n^{1+1/k})$ edges [6, 35, 47]. This is known to be tight for $k = 1, 2, 3, 5$ and is conjectured to be tight for all k (see, for example a survey by Zwick [50]). Undirected spanners have numerous applications, such as efficient routing [15, 16, 38, 39, 46],

*All omitted proofs and details appear in the full version [10].

†Massachusetts Institute of Technology, USA. Email: {abhatt, elena_g,kmjung}@mit.edu. A.B. was supported by National Science Foundation (NSF grants 0514771, 0732334, and 0728645) and DOE Computational Science Graduate Fellowship. E.G. was funded in part by NSF grants CCR-0726525 and CCR-0829672.

‡Pennsylvania State University, USA. Email: sofya@cse.psu.edu. Supported by National Science Foundation (NSF grant CCF-0729171).

§IBM Almaden Research Center, USA. Email: dpwoodru@us.ibm.com.

simulating synchronized protocols in unsynchronized networks [37], parallel and distributed algorithms for approximating shortest paths [13, 14, 19], and algorithms for distance oracles [9, 47].

In the directed setting, two notions of spanners have been considered in the literature: the direct generalization of the above definition [36] and *roundtrip spanners* [16, 39]. In this paper, we introduce a new definition of directed spanners that captures the notion that a spanner should have a small diameter but preserve the connectivity of the original graph.

DEFINITION 1.1. (TC-SPANNER) *Given a directed graph $G = (V, E)$ and an integer $k \geq 1$, a k-**transitive-closure-spanner** (k-**TC-spanner**) is a directed graph $H = (V, E_H)$ with the following properties: (1) E_H is a subset of the edges in the transitive closure of G. (2) For all vertices $u, v \in V$, if $d_G(u, v) < \infty$, then $d_H(u, v) \leq k$.*

Notice that a k-TC-spanner of G is just a directed k-spanner of the transitive-closure of G. Nevertheless, TC-spanners are interesting in their own right due to the numerous TC-spanner-specific applications we present in Section 1.3.

One of the focuses of this paper is the study of the computational problem of finding the size of the sparsest k-TC-spanner for a given digraph, referred to as k-TC-SPANNER. It is a special case of the problem of finding the size of the sparsest directed spanner, called DIRECTED k-SPANNER, that has been previously studied. Both problems are NP-hard (proofs appear in the full version [10]).

1.1 Related Work Thorup [42] considered a special case of TC-spanners of graphs G that have at most twice as many edges as G, and conjectured that for all directed graphs G on n nodes there are such TC-spanners with stretch polylogarithmic in n. He proved his conjecture for planar graphs [43], but later Hesse [30] gave a counterexample to Thorup's conjecture for general graphs. TC-spanners were also studied for directed trees: implicitly in [5, 8, 11, 17, 49] and explicitly in [44]. For the directed line, [5] (and later, [8]) showed that the size of the sparsest k-TC-spanner is $\Theta(n \cdot \lambda_k(n))$, where $\lambda_k(n)$ is the k^{th}-row inverse Ackermann function. [5, 11, 44] gave the same bounds for directed trees.

Approximability of directed spanner problems. All algorithms for DIRECTED k-SPANNER immediately yield algorithms for k-TC-SPANNER with the same approximation ratio. Kortsarz and Peleg [33] give an $O(\log n)$-approximation algorithm for DIRECTED-2-SPANNER, and Kortsarz [32] shows that this approximation ratio cannot be improved unless P=NP. For $k = 3$, Elkin and Peleg [20] present an $\tilde{O}(n^{2/3})$-approximation algorithm. Their algorithm is complicated, and the $\mathrm{polylog}$ factor hidden in the $\tilde{O}$ notation is not analyzed. For $k \geq 4$, sublinear factor approximation algorithms are known only in the undirected setting [36]. We note that Dodis and Khanna [18] and Chekuri *et al.* [12] study algorithms that might seem relevant to k-TC-SPANNER. In [10], we explain why these algorithms do not work for k-TC-SPANNER.

For all constant $k > 2$ and $\epsilon \in (0, 1)$, it is impossible to approximate DIRECTED k-SPANNER within a factor of $2^{\log^{1-\epsilon} n}$, assuming NP$\not\subseteq$DTIME($n^{poly \log n}$) [20]. Moreover, [23] extend this result to $3 \leq k = O(n^{1-\delta})$ for all $\delta \in (0, 1)$. Thus, according to Arora and Lund's classification [31] of NP-hard problems, DIRECTED k-SPANNER is in class III, for $3 \leq k = O(n^{1-\delta})$. Moreover, [23] show that proving that DIRECTED k-SPANNER is in class IV, that is, inapproximable within n^{δ} for some $\delta \in (0, 1)$, would resolve a long standing open question in complexity theory, and cause classes III and IV to collapse into a single class.

1.2 Our Contributions In this work we (1) bring several diverse applications, including property testing, access control and data structures, under the unifying framework of TC-spanners, (2) obtain bounds on the approximability of k-TC-SPANNER, DIRECTED k-SPANNER and well-studied variants of these problems, and (3) construct sparse TC-spanners for the family of H-minor free graphs, which include planar, bounded-treewidth, and bounded genus graphs. Table 1 summarizes our results on the approximability of k-TC-SPANNER.

Algorithms for k-TC-SPANNER and related problems. We present two deterministic polynomial time approximation algorithms for k-TC-SPANNER. Our first algorithm uses a new combination of convex programming and sampling, and gives an $O((n \log n)^{1-1/k})$-ratio for k-TC-SPANNER. Moreover, our method yields the same approximation ratio for DIRECTED k-SPANNER and its well-studied variants: CLIENT/SERVER DIRECTED k-SPANNER, and k-DIAMETER SPANNING SUBGRAPH (see [21] for definitions). This resolves the open question of finding a sublinear approximation ratio for these problems for $k > 3$, described as a "challenging direction" for research on directed spanners by Elkin and Peleg [22]. Our algorithm for $k = 3$ is arguably simpler than the $O(n^{2/3} \,\mathrm{polylog}\, n)$-approximation algorithm of [22].

Our second algorithm has an $\tilde{O}(n/k^2)$ ratio for k-TC-SPANNER. This demonstrates a separation between k-TC-SPANNER and DIRECTED k-SPANNER: for $k = \sqrt{n}$, it gives $O(\log n)$-approximation for k-TC-SPANNER while [23, Theorem 6.6] showed that DIRECTED $\sqrt{n}$-SPANNER is $2^{\log^{1-\epsilon} n}$-inapproximable. Moreover, Hesse [30] asks for an algorithm to add $O(|G|)$ "shortcuts" to a digraph and reduce its diameter to $\sqrt{n}$. Our second algorithm returns a $\sqrt{n}$-TC-spanner of size $O(|G| + \log n)$, answering his question.

Inapproximability of k-TC-SPANNER. We present two results on the hardness of k-TC-SPANNER. First, we prove for $k = 2$ that the $O(\log n)$ ratio of [33] is optimal unless P=NP. Next, for constant $k > 2$, we show that k-TC-

Setting of k	Implied by previous work	This paper	Notes
$k = 2$	$O(\log n)$ [33]	$\Omega(\log n)$	
constant $k > 2$		$\Omega(2^{\log^{1-\epsilon} n})$	
$k = 3$	$O(n^{2/3} \operatorname{polylog} n)$ [20]	$O((n \log n)^{2/3})$	applies to DIRECTED k-SPANNER
$k > 3$	$O(n)$ [trivial]	$O((n \log n)^{1-1/k})$	
$k = \Omega\left(\frac{\log n}{\log \log n}\right)$	$O(n)$ [trivial]	$O\left(\frac{n \log n}{k^2 + k \log n}\right)$	separation from DIRECTED k-SPANNER

Table 1: Summary of Results on Approximability of k-TC-SPANNER

SPANNER is inapproximable within a factor of $2^{\log^{1-\epsilon} n}$, for all $\epsilon \in (0, 1)$, unless NP$\subseteq$DTIME($n^{\operatorname{polylog} n}$). This result is our main technical contribution. Observe that a stronger inapproximability result for $k > 2$ would imply the same inaproximability for DIRECTED-k-SPANNER, and as shown in [23], collapse classes III and IV in Arora and Lund's classification.

Our $2^{\log^{1-\epsilon} n}$-hardness matches the known hardness for DIRECTED k-SPANNER. As is the case for DIRECTED k-SPANNER, we start by building a directed graph from a well-known hard problem called MIN-REP, which has the same inapproximability as SYMMETRIC LABEL COVER. However, as illustrated in Section 3, all known hard instances for DIRECTED k-SPANNER cannot imply anything better than $\Omega(1)$-hardness for k-TC-SPANNER. Intuitively, our lower bound is much harder to prove than the one for DIRECTED k-SPANNER since our instance must be transitively-closed, and thus, many more "shortcut" routes between pairs of vertices exist. Our construction uses a novel application of the generalized butterfly and broom graphs, together with several transformations of the MIN-REP problem, which make it *noise-resilient*. We call a MIN-REP instance noise-resilient to indicate that its structure is preserved under small perturbations. The paths in the generalized butterfly are well-structured, which allows us to analyze the many different routes possible in the transitive closure.

Structural results. Finally, we study the minimum k-TC-spanner size for a specific graph family with sparse k-TC-spanners: H-minor-free graphs. A graph H is a *minor* of G if H is a subgraph of a graph obtained from G by a sequence of edge contractions and deletions. For a fixed graph H (e.g., K_5), the family of *H-minor-free graphs* is a minor-closed family that excludes H. Examples of such families include planar graphs, bounded treewidth graphs, and bounded genus graphs, explicitly studied in applications in Section 1.3. For H-minor-free graphs, we efficiently construct 2-TC-spanners of size $O(n \log^2 n)$, and k-TC-spanners of size $O(n \cdot \log n \cdot \lambda_k(n))$, where $\lambda_k(\cdot)$ is the k^{th}-row inverse Ackermann function. The main idea is to use the path separators for undirected H-minor free graphs due to Abraham and Gavoille [1]. However, although the separators are paths, in our digraph they may be the union of many dipaths, and so we cannot efficiently recurse using the sparse k-TC-spanners for the directed line of Alon and Schieber [5]. We observe that these separators satisfy a stronger property than claimed in [1], effectively allowing us to encode the direction of edges in a cost function associated with the separators.

1.3 Applications of TC-spanners

Monotonicity testing. Monotonicity of functions [4, 17, 24, 25, 26, 27, 29] is one of the most studied properties in property testing [28, 40]. Fischer *et al.* [26] prove that testing monotonicity is equivalent to several other testing problems. Let V_n be a poset of n elements and $G_n = (V_n, E)$ be the relation graph, i.e., the Hasse diagram, for V_n. A function $f : V_n \to \mathbb{R}$ is called *monotone* if $f(x) \leq f(y)$ for all $(x, y) \in E$. We say f is ϵ-far from monotone if f has to be changed on $\geq \epsilon$ fraction of the domain to become monotone, that is, $\min_{\text{monotone } g} |\{x : f(x) \neq g(x)\}| \geq \epsilon n$. A monotonicity tester on G_n is an algorithm that, given an oracle for a function $f : V_n \to \mathbb{R}$, passes if f is monotone but fails with probability $\geq \frac{2}{3}$ if f is *ϵ-far* from monotone. The optimal monotonicity tester for the directed line L_n, consisting of nodes $\{1, 2, \ldots, n\}$ and edges $\{(i, i+1) : 1 \leq i \leq n-1\}$, proposed by Dodis *et al.* [17], is based on the sparsest 2-TC-spanner for that graph. Implicit in the proof of Proposition 9 in [17] is a lemma relating the complexity of a monotonicity tester for L_n to the size of a 2-TC-spanner for L_n. We generalize this by observing that a sparse 2-TC-spanner for any partial order graph G_n implies an efficient monotonicity tester on G_n.

LEMMA 1.1. *If a directed acyclic graph G_n has a 2-TC-spanner with $s(n)$ edges, then there exists a monotonicity tester on G_n that runs in time $O\left(\frac{s(n)}{\epsilon n}\right)$.*

Proof. The tester selects $\frac{8s(n)}{\epsilon n}$ edges of the 2-TC-spanner H uniformly at random. It queries function f on the endpoints of all the selected edges and rejects if some selected edge (x, y) is *violated* by f, that is, $f(x) > f(y)$.

If the function f is monotone on G_n, the algorithm always accepts. The crux of the proof is to show that functions that are ϵ-far from monotone are rejected with

probability at least $\frac{2}{3}$. Let $f : V_n \to \mathbb{R}$ be a function that is ϵ-far from monotone. It is enough to demonstrate that f violates at least $\frac{\epsilon n}{4}$ edges in H. Then each selected edge is violated with probability $\frac{\epsilon n}{4s(n)}$, and the lemma follows by elementary probability theory.

Denote the transitive closure of G by $TC(G)$. We say a vertex $x \in V_n$ is assigned a *bad* label by f if x has an incident violated edge in $TC(G_n)$; otherwise, x has a *good* label. Let V' be a set of vertices with good labels. Observe that f is monotone on the induced subgraph $G' = (V', E')$ of $TC(G)$. This implies ([26], Lemma 1) that f can be changed into a monotone function by modifying it on at most $|V_n - V'|$ vertices. Since f is ϵ-far from monotone, it shows that there are at least ϵn vertices with bad labels.

Every function that is ϵ-far from monotone has a matching M of at least $\frac{\epsilon n}{2}$ violated edges in $TC(G)$ [17]. We will establish a map from the set of edges in M to the set of violated edges in H, so that each violated edge in H is the image of at most 2 edges in M. For each edge (x, y) in the matching, consider the corresponding path from x to y of length at most 2 in the 2-TC-spanner H. If the path is of length 1, (x, y) is the violated edge in H corresponding to the matching edge (x, y). Otherwise, let (x, z, y) be a path of length 2 in H. At least one of the edges (x, z) and (z, y) is violated, and we map (x, y) to that edge. Since M is a matching, at most 2 edges in M can be mapped to one violated edge in $TC(G)$. Thus, the 2-TC-spanner H has $\geq \frac{\epsilon n}{4}$ violated edges, as required. □

Therefore, all the 2-TC-spanner constructions described in this paper yield monotonicity testers for functions defined on the corresponding posets. Moreover, for H-minor free graphs, the resulting tester has much better query complexity than the previously known, due to Fischer *et al.* [26]. Indeed, we achieve testers with $O(\log^2 n/\epsilon)$ queries, whereas previous testers required $\Theta(\sqrt{n}/\epsilon)$ queries.

Key management in an access hierarchy. In the problem of key management in an access hierarchy, i.e., access control, there is a partially ordered set (poset) of access classes and a key associated with each class. This is modeled by a directed graph G whose nodes are classes and whose edges indicate an ordering. A user is entitled to access a certain class and all classes reachable from it. This problem arises in content distribution, operating systems, and project development (see, e.g., the references in [8]). One approach to the access control problem [7, 8, 41] is to associate public information $P(i, j)$ with each edge $(i, j) \in G$ and a secret key k_i with each node i. There is an efficient algorithm A which takes k_i and $P(i, j)$ and generates k_j. However, for each (i, j) in G, it is computationally hard to generate k_j without knowledge of k_i. To obtain a key k_v from a key k_u, algorithm A is run $d_G(u, v)$ times. To speed this up, [8] suggest adding edges to G to increase connectivity. To preserve the access hierarchy of G, new edges must be from the transitive closure of G. The number of edges added corresponds to the space complexity of the scheme, while the shortest-path distances correspond to the time complexity. Implicit in [8] are TC-spanners for directed trees with $k = 3$ and size $O(n \log\log n)$ and also with $k = O(\log\log n)$ and size $O(n)$. Our results for H-minor free graphs extend the known posets for which access control schemes have $O(n \operatorname{polylog} n)$ storage and $O(1)$ key derivation time. Our approximation algorithms yield sparse k-TC-spanners for general posets.

Partial products in a semigroup. Yao [49] and Alon and Schieber [5] study space-efficient data structures for the following problem: Preprocess elements $\{s_1, \ldots, s_n\}$ of a semigroup $(S, \circ)$, such as $(\mathbb{R}, \min)$, to be able to compute partial products $s_i \circ s_{i+1} \circ \cdots \circ s_j$ for all $1 \leq i < j \leq n$ with at most k queries to a small database of pre-computed partial products. This problem reduces to finding a sparsest k-TC-spanner for a directed line L_{n+1}. Chazelle [11] and Alon and Schieber also consider a generalization of the above problem, where the input is an (undirected) tree T with an element s_i of a semigroup associated with each vertex i. The goal is to create a space-efficient data structure that allows one to compute the product of elements associated with all vertices on the path from i to j, for all vertex pairs i, j in T. The generalized problem reduces to finding a sparsest k-TC-spanner for a directed tree T' obtained from T. We describe the reduction in the full version of this paper [10].

Organization. Section 2 contains an overview of our algorithms. In Section 3, we give an overview of our lower bounds and the techniques involved. Section 4 contains an overview of our bounds for minor-free graphs. We defer the details and proofs of our results to [10].

Notation. The *transitive closure* of a graph $G = (V, E)$, denoted $TC(G)$, is the directed graph (V, E'), where $E' = \{(u, v) : u \rightsquigarrow_G v\}$. Vertices u and v are *comparable* if either $(u, v) \in TC(G)$ or $(v, u) \in TC(G)$. The *transitive reduction* of G, denoted $TR(G)$, is a digraph G' with the fewest edges for which $TC(G') = TC(G)$. As shown by Aho *et al.* [3], $TR(G)$ can be computed efficiently via a greedy algorithm. For directed acyclic graphs $TR(G)$ is unique, and G is *transitively reduced* if $TR(G) = G$. We call an edge a *shortcut edge* if it is in $TC(G)$ but not in G.

The *Ackermann function* [2] is defined by: $A(1, j) = 2^j$, $A(i+1, 0) = A(i, 1)$, $A(i+1, j+1) = A(i, 2^{2^{A(i+1,j)}})$. The inverse Ackermann function is $\alpha(n) = \min\{i : A(i, 1) \geq n\}$ and the i^{th}-row inverse is $\lambda_i(n) = \min\{j : A(i, j) \geq n\}$.

2 Overview of Algorithms for k-TC-Spanner and Related Problems

Our $O((n \log n)^{1-1/k})$-approximation for k-TC-Spanner for arbitrary k is based on a new combination of convex programming and sampling. Our technique also achieves an $O((n \log n)^{1-1/k})$ ratio for Directed k-

SPANNER, CLIENT/SERVER DIRECTED k-SPANNER, and k-DIAMETER SPANNING SUBGRAPH. Here we describe the result for DIRECTED k-SPANNER. To achieve the same result for k-TC-SPANNER, it suffices to run the algorithm on the transitive-closure of the input digraph.

THEOREM 2.1. *For any (not necessarily constant) $k > 2$, there is a deterministic polynomial-time algorithm achieving an $O((n \log n)^{1-1/k})$-approximation for* DIRECTED k-SPANNER.

We start by formulating the problem as an integer program. We briefly explain the problems with this approach and the ideas required to make it work. One can introduce binary edge variables x_e for each edge e in the transitive closure, and binary path variables y_P for each path P of length $\leq k$ in the transitive closure. One enforces the constraints $y_P \leq x_e$ for each $e \in P$, which allow a path P in the spanner only if all edges along it are present. The final constraint is $\sum_P y_p \geq 1$ for all edges $(u, v) \in G$, where the sum is over paths P of length $\leq k$ from u to v. Finally, one can relax the problem to an LP, and try to round the solution.

The first problem is that the integrality gap is huge, which may be why an LP approach had not been considered before. Indeed, if there are $\Theta(n)$ paths of length at most k (say, for constant k) between u and v, the LP might assign each of them a value of $\Theta(1/n)$. However, we observe that if there are $r = n^{1-1/k}$ distinct paths from u to v of length $\leq k$, there must be $\geq r^{1/(k-1)}$ distinct vertices w for which $u \rightsquigarrow w \rightsquigarrow v$. Let $BFS(v)$ denote a shortest path tree of edges directed away from v, together with a shortest path tree of edges directed towards v. We sample $\tilde{O}(n/r^{1/(k-1)})$ vertices, and grow $BFS(w)$ of $2(n-1)$ edges around each sample w. Then we are likely to sample a w for which $u \rightsquigarrow w \rightsquigarrow v$, and the path from u to v along the edges in $BFS(w)$ has length $\leq k$. We let the spanner H be the union of the outputs of the LP and sampling-based algorithms.

1. $H \leftarrow \emptyset$.
2. For each edge $e \in G$, if $x_e \geq \frac{1/2}{(n \log n)^{1-1/k}}$, $H \leftarrow H \cup \{e\}$.
3. Randomly sample $r = O((n \log n)^{1-1/k})$ vertices $z_1, z_2, \ldots, z_r \in G$.
4. $H \leftarrow H \cup (\cup_i BFS(z_i))$. Output H.

With high probability, an edge (u, v) is covered by either the LP relaxation or the sampling.

LEMMA 2.1. *With probability at least $1 - 1/n$, H is a k-TC-spanner of G.*

The spanner has at most $r \cdot OPT + \frac{n^2}{r^{1/(k-1)}}$ edges, where OPT is the optimum of the LP. By observing that any spanner must have size $\min(OPT, n-1)$, one can guarantee that this is an $\tilde{O}(n^{1-1/k})$-approximation. Note that we assume that G is connected, as otherwise we can run the algorithm separately on each component. A more careful analysis gives an $O((n \log n)^{1-1/k})$-approximation, and a simple greedy algorithm derandomizes the sampling.

LEMMA 2.2. $|H| = O((n \log n)^{1-1/k} OPT)$.

The problem with this approach is that the number of variables and the size of each of the constraints grows exponentially with k. We replace the variables y_P with $\min_{e \in P} x_e$, reducing the number of variables to $O(n^2)$. The resulting program is convex, and we use the ellipsoid algorithm with a separation oracle. The oracle, given $\vec{x}$, just needs to find one pair of vertices (u, v) for which the constraint $\sum_{P: u \rightsquigarrow v} \min_{e \in P} x_e \geq 1$ is violated. It can do this by sorting the coordinates of $\vec{x}$, and counting the number of u-v paths P for which some particular x_e is the minimum edge variable along P. For this, it iteratively removes edges e from G for which x_e is smallest, and uses matrix multiplication to count the u-v paths that remain in the graph.

LEMMA 2.3. *For any k, there exists a separation oracle which runs in time* $\mathrm{poly}(n)$.

k-TC-SPANNER algorithm for large k. Our $\tilde{O}(n/k^2)$-approximation algorithm, which is specific to k-TC-SPANNER, works by sampling $\tilde{O}(n/k)$ vertices and selecting $O(n/k)$ edges from the transitive closure adjacent to the samples. We also include the edges of $TR(G)$ in the spanner. A simple greedy algorithm derandomizes the sampling.

THEOREM 2.2. *For any k, there exists a deterministic approximation algorithm for the k-TC-SPANNER problem with approximation ratio $O((n \log n)/(k^2 + k \log n))$.*

3 Overview of Hardness Results for k-TC-Spanner

This section outlines the proof of Theorem 3.1, which is our main technical contribution. Missing details appear in [10]. At the end we briefly describe the ideas behind the inapproximability result for 2-TC-SPANNER.

THEOREM 3.1. *For any fixed $\epsilon \in (0, 1)$, the size of the sparsest k-TC-spanner cannot be approximated to within a factor of $2^{\log^{1-\epsilon} n}$ unless $NP \subseteq DTIME(n^{\mathrm{polylog}\, n})$.*

3.1 The Construction and its Motivation Since k-TC-SPANNER is a special case of DIRECTED k-SPANNER, which is $\Theta(\log n)$-inapproximable for $k = 2$ and $2^{\log^{1-\epsilon} n}$-inapproximable for $k \geq 3$, it is natural to ask whether the hard instances of DIRECTED k-SPANNER from [32, 20, 23] can be used to prove hardness for k-TC-SPANNER. It turns out that all these instances have very small k-TC-spanners. We demonstrate it for the instance used in the

proof of $\Omega(\log n)$-hardness for DIRECTED k-SPANNER, which works via a reduction from SET-COVER.

Let G be a bipartite digraph for SET-COVER with n vertices ("sets") on the left, n vertices ("elements") on the right, and edges from left to right. Let I be a set of i new independent vertices, for some value i, and let L be a directed line on $k-1$ new vertices. Call the first vertex of L the head, and the last vertex the tail. Include directed edges (1) from the tail of L to every set in G, (2) from every vertex of I to the head of L, and (3) from every vertex of I to the sets and the elements of G. Call the constructed digraph G'.

Observe that in G', all directed edges except those from I to G must be included in the directed k-spanner, as such edges form the unique path between their endpoints. At this point, the only pairs of vertices at distance larger than k are those from a vertex in I to an element of G. Since these vertices are adjacent in G', there must be a path of length at most k in the spanner. The only possible path is from the vertex in I to a vertex of G. It is easy to see that adding exactly OPT edges from each vertex in I to the sets of G is necessary and sufficient to obtain a spanner, where OPT is the size of the minimum set-cover. By making i sufficiently large, the size of the spanner is easily seen to be $\Theta(i \cdot OPT)$, and thus one can approximate SET-COVER by approximating DIRECTED k-SPANNER, so the problem is $\Omega(\log n)$-inapproximable.

However, there is a trivial k-TC-spanner for this instance! Indeed, by transitivity we can simply connect the head of L to each of the elements of G. This is a k-TC-spanner of size proportional to the number of vertices in G'. Thus, the best one could hope for with this instance is to show $\Omega(1)$-hardness for k-TC-SPANNER. For similar reasons, the instance showing $2^{\log^{1-\epsilon} n}$-inapproximability for DIRECTED k-SPANNER also cannot establish anything beyond $\Omega(1)$-hardness for k-TC-SPANNER.

In the example above there are many paths to cover (those from I to elements of G), but a few "shortcut" edges cover them all. Ideally, we would have many paths to cover, and each shortcut edge could only cover a single path. Hesse's digraph requiring a large number of shortcuts to reduce its diameter [30] satisfies the desired condition. His idea was to associate vertices with a subset V of vectors in $\mathbb{R}^d$ such that $(u, v) \in E$ iff $u - v$ is an extreme point of the d-dimensional ball of integer points. By the properties of an extreme point, a shortcut can cover at most one path from a large family of shortest paths.

However, to achieve an inapproximability result, we need better structured graphs. We use *generalized butterflies* defined in [48]. In these digraphs vertices are identified with coordinates $[n^{1/k}]^k \times [k+1]$, and an edge connects $u = (u_1, \ldots, u_k, i)$ to $v = (v_1, \ldots, v_k, i+1)$ iff for all $j \neq i$, $u_j = v_j$. We say a vertex $(u_1, \ldots, u_k, i)$ is in *strip* i. It is easy to see that there is a unique shortest path of length k from any u in strip 1 to any v in strip $k+1$. Moreover, any shortcut is on at most $n^{1-2/k}$ such paths because if it connects a vertex in strip i with a vertex in strip $i+\ell$ (where $\ell \geq 2$) it fixes all but $i-1$ coordinates of u and all but $k+1-(i+\ell)$ coordinates of v. Thus, $\geq n^{1+2/k}$ shortcuts are needed to reduce the diameter to $k-1$.

Reduction from MIN-REP**.** To get $2^{\log^{1-\epsilon} n}$- inapproximability, we reduce from the MIN-REP problem. An (n, r, d, m)-MIN-REP instance is a bipartite graph of maximum degree d in which the left part can be partitioned into sets $\mathcal{A}_1, \ldots, \mathcal{A}_r$ and the right part into sets $\mathcal{B}_1, \ldots, \mathcal{B}_r$, so that $|\mathcal{A}_i| = |\mathcal{B}_i| = n/r$ for all $i \in [r]$. To describe the last parameter m, call a vertex *isolated* if its degree is 0, and *non-isolated* otherwise. Let $m(\mathcal{A}_i)$ be the inverse of the fraction of non-isolated vertices in $\mathcal{A}_i$. Then m is the minimum such $m(\mathcal{A}_i)$. Define the *supergraph* to have nodes $\mathcal{A}_1, \ldots, \mathcal{A}_r, \mathcal{B}_1, \ldots, \mathcal{B}_r$, with a *superedge* $(\mathcal{A}_i, \mathcal{B}_j)$ iff there is a node in $\mathcal{A}_i$ adjacent to a node in $\mathcal{B}_j$. A *rep-cover* is a vertex set S in the graph such that whenever $(\mathcal{A}_i, \mathcal{B}_j)$ is an edge in the supergraph, there is an edge between some $u, v \in S$ with $u \in \mathcal{A}_i$ and $v \in \mathcal{B}_j$. A solution to MIN-REP is a smallest rep-cover, and its size is denoted by OPT. The problem is $2^{\log^{1-\epsilon} n}$-inapproximable [20].

As a first attempt, we construct a graph G of diameter $k+2$ as follows. We attach a disjoint copy of a generalized butterfly of diameter $k-1$ to each $\mathcal{A}_i$ in the MIN-REP instance graph; that is, we identify the vertices in $\mathcal{A}_i$ with the last strip of the butterfly. We call the vertices in the butterfly at distance x from $\mathcal{A}_i$ the x-th *shadow* of $\mathcal{A}_i$. Next, for each $\mathcal{B}_j$, we attach what we call a *broom*. This is a 3-layer graph, where the two leftmost layers form a bipartite clique, and the right layer consists of degree-1 nodes, called *broomsticks*, attached to nodes in the middle layer. Each node in the middle layer has the same number of broomsticks attached to it. Each $\mathcal{B}_j$ is identified with the left layer of a disjoint broom. All edges of G are directed from the shadows of the $\mathcal{A}_i$ towards the broomsticks (left to right).

We would like to argue that the minimum k-TC-spanner H of G is formed as follows. Let S be a minimum rep-cover of the underlying MIN-REP instance. For each $s \in S$, if s is in an $\mathcal{A}_i$, include all shortcuts from the 2-shadow of $\mathcal{A}_i$ to s which are in the transitive closure of G. Otherwise (s is in a $\mathcal{B}_j$), include all shortcuts from s to the broomsticks of $\mathcal{B}_j$. By balancing the number of broomsticks with the size of 2-shadows, one can show H has size $|S|f(n,k)$, where $f(n,k)$ is an easily computable function. Since S is a rep-cover, H is a k-TC-spanner. If H were optimal, then approximating its size within some factor would approximate MIN-REP within the same factor.

It turns out that H is not optimal, and so our first attempt does not work. Below, we modify G and consider a related k-TC-spanner H of the modified G. We show that any k-TC-spanner has size $\Omega(|H|/\log n)$ for constant k. Since MIN-

REP is $2^{\log^{1-\epsilon} n}$-inapproximable, this still gives $2^{\log^{1-\epsilon} n}$-hardness.

To prove this, we need to argue that most vertices v in the k-shadows do not "benefit" from traversing other shortcuts to reach the broomsticks. This requires a classification of all alternative routes from such v to broomsticks. Sine v is in a generalized butterfly, these routes are well-understood. However, for a generic MIN-REP instance, most of these routes do indeed lead to a much smaller k-TC-spanner.

To rule out the alternative routes, we ensure that OPT and the four parameters of the MIN-REP instance each lie in a narrow range. In Theorem 3.2, we prove that MIN-REP with the required parameter restrictions is inapproximable by giving a reduction from an unrestricted MIN-REP instance. It works by carefully interleaving the following five operations on a "base" MIN-REP instance with unrestricted parameters: (1) disjoint copies, (2) dummy vertices inside clusters, (3) blowup inside clusters with matching supergraph, (4) blowup inside clusters with complete supergraph, and (5) tensoring. Each operation increases one or several parameters by a prespecified factor, and together they give us five degrees of freedom to control the range of OPT and the four parameters of MIN-REP.

THEOREM 3.2. (**Noise-Resilient MIN-REP is hard)** *Fix parameters $\kappa \in (0,1)$ and $R, D, M, F \in (0, 1-\kappa)$ satisfying $F \in (R, 2R)$ and $D + M + F < 1$.* Noise-Resilient MIN-REP *is a family of (n, r, d, m)*-MIN-REP *instances with $r \in [n^R, n^{R+\kappa}]$, $d \in [n^D, n^{D+\kappa}]$, $m \in [n^M, n^{M+\kappa}]$, and $OPT \in [n^F, n^{F+\kappa}]$. This problem is $2^{\log^{1-\epsilon} n}$-inapproximable for all $\epsilon \in (0,1)$ unless $NP \subseteq DTIME(n^{\mathrm{polylog}\, n})$.*

The variant of MIN-REP in Theorem 3.2 is called "noise-resilient" because even if many vertices in the sets $\mathcal{A}_i$ and $\mathcal{B}_j$ are adversarially deleted in an instance of this problem, the minimum rep-cover does not shrink significantly. This property helps us rule out many alternative routes in the TC-spanner, though we will need to change our graph G. Our reduction from noise-resilient MIN-REP to k-TC-SPANNER for $k > 2$ consists of two steps: first we produce a *specialized* MIN-REP instance $\mathcal{I}$ from an arbitrary instance $\mathcal{I}_0$ of noise-resilient MIN-REP, and then we construct a k-TC-SPANNER instance $\mathcal{G}$ by carefully adjoining generalized butterflies on the left and broom graphs on the right of $\mathcal{I}$.

From noise-resilient MIN-REP **to specialized** MIN-REP**.** Set $\delta = \frac{k-1}{k-\frac{1}{4}}$, $\eta = \frac{\delta}{2(4k-4)(4k-2)}$, and $\zeta = \delta\left(\frac{4k-5}{4k-4} + \frac{1}{4k-2}\right)$. Let κ be a sufficiently small positive constant. We start from an (n_0, r_0, d_0, m_0)-instance $\mathcal{I}_0$ of noise-resilient MIN-REP with optimum OPT_0, where $n_0 = n^\delta, r_0 \in [n^{\delta/2}, n^{\delta/2+\kappa}], d_0 \in [n^\eta, n^{\eta+\kappa}]$, $m_0 \in [n^{2\eta}, n^{2\eta+\kappa}]$, and $OPT_0 \in [n^\zeta, n^{\zeta+\kappa}]$. By instantiating Theorem 3.2 with $R = \frac{1}{2}, D = \frac{\eta}{\delta}, M = \frac{2\eta}{\delta}$, $F = \frac{\zeta}{\delta}$ and κ, we obtain that the (n_0, r_0, d_0, m_0)-MIN-REP problem is $2^{\log^{1-\epsilon} n}$-inapproximable unless $NP \subseteq DTIME(n^{\mathrm{polylog}\, n})$. The conditions on the parameters in Theorem 3.2 are satisfied since $\zeta \in (\frac{\delta}{2}, \delta)$ and $\eta + 2\eta + \zeta < \delta$.

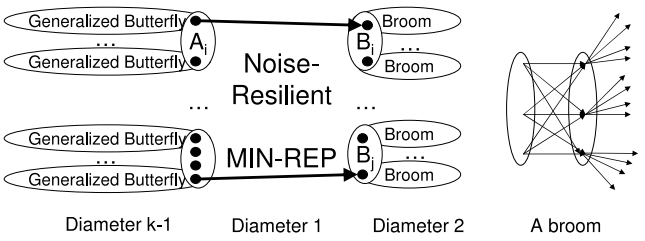

Figure 1: TC-spanner instance $\mathcal{G}$ & an example of a broom.

We transform $\mathcal{I}_0$ to a specialized (n, r, d, m)-MIN-REP instance $\mathcal{I}$ with $r = r_0$, $d = d_0 n^{1-\delta}$ and $m = m_0$. Graph $\mathcal{I}$ is bipartite, with nodes partitioned into clusters $\mathcal{A}_1, \ldots, \mathcal{A}_r$ on the left, and $\mathcal{B}_1, \ldots, \mathcal{B}_r$ on the right. Each $\mathcal{A}_i$ and $\mathcal{B}_j$ is a union of $n^{1-\delta}$ *groups* $A_{i,s}$ and $B_{j,s}$, respectively, with $s \in [n^{1-\delta}]$. Each group $A_{i,s}$ and $B_{j,s}$, for $i, j \in [r]$, $s \in [n^{1-\delta}]$, is a copy of $\mathcal{A}_i$ and, respectively, $\mathcal{B}_j$, from the original instance $\mathcal{I}_0$. For each edge (u, v) with $u \in \mathcal{A}_i$ and $v \in \mathcal{B}_j$ of $\mathcal{I}_0$, graph $\mathcal{I}$ has edges between the copy of u in A_{i,k_1} and the copy of v in B_{j,k_2}, for all $k_1, k_2 \in [n^{1-\delta}]$. The solution value of $\mathcal{I}$ remains OPT_0 because the supergraph corresponding to $\mathcal{I}_0$ and $\mathcal{I}$ are identical.

From specialized MIN-REP **to** k-TC-SPANNER**.** From $\mathcal{I}$, we construct a graph $\mathcal{G}$ of diameter $k+2$ as follows. We first attach a disjoint generalized butterfly of diameter $k-1$, denoted $BF(A_{i,s})$, to each group $A_{i,s}$ in $\mathcal{I}$, for all $i \in [r]$, $s \in [n^{1-\delta}]$. That is, we identify vertices in $A_{i,s}$ with the last strip of $BF(A_{i,s})$ in the way discussed below. Denote by $BF(\mathcal{A}_i) = \cup_s BF(A_{i,s})$ the set of all the vertices attached in this manner to the cluster $\mathcal{A}_i$. Let $BF^j(A_{i,s})$ be the vertices in strip j of the butterfly $BF(A_{i,s})$, where $BF^k(A_{i,s}) = A_{i,s}$, and let $BF^j(\mathcal{A}_i) = \cup_s BF^j(A_{i,s})$. We call the vertices in the butterfly $BF(A_{i,s})$ at distance x from $A_{i,s}$ the x-th *shadow* of $A_{i,s}$. Call the in-degree as well as out-degree of the vertices in the butterflies $d_* \stackrel{\text{def}}{=} (\frac{n^\delta}{r})^{\frac{1}{k-1}}$.

Next, for each $\mathcal{B}_{i,s}$, we attach a broom, denoted $BR(B_{i,s})$. More specifically, each vertex in $B_{i,s}$ is connected to the vertices of a set $BR^{k+2}(B_{i,s})$ of size d_*, and each vertex $v \in BR^{k+2}(B_{i,s})$ is connected to a disjoint set of nodes, called broomsticks, of size d_*. Let $BR^{k+3}(B_{i,s})$ be the set of broomsticks adjacent to $BR^{k+2}(B_{i,s})$. Let $BR^{k+2}(\mathcal{B}_i) = \cup_s BR^{k+2}(B_{i,s})$ and $BR^{k+3}(\mathcal{B}_i) = \cup_s BR^{k+3}(B_{i,s})$. Identify *layer* V_j with $\cup_{i,s} BF^j(A_{i,s})$ for $j \in [k]$, layer V_{k+1} with $\cup_{i,s} B_{i,s}$, and layer V_j with $\cup_i BR^j(\mathcal{B}_i)$ for $j \in \{k+2, k+3\}$. Direct all the edges from V_i to V_{i+1}. See Figure 1.

Attaching butterflies. Recall that we identify vertices in $A_{i,s}$ with the last strip $BF^k(A_{i,s})$ of a disjoint butterfly,