SIAM J. COMPUT.
Vol. 41, No. 6, pp. 1380–1425

© 2012 Society for Industrial and Applied Mathematics

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TRANSITIVE-CLOSURE SPANNERS*

ARNAB BHATTACHARYYA[†], ELENA GRIGORESCU[‡], KYOMIN JUNG[§], SOFYA RASKHODNIKOVA[¶], AND DAVID P. WOODRUFF[‖]

Abstract. 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 implicitly studied in the context of circuit complexity, data structures, property testing, and access control, and properties of these spanners have been rediscovered over the span of 20 years. We abstract the common task implicitly tackled in these diverse applications as the problem of constructing sparse TC-spanners. We initiate the study of approximability of the size of the sparsest k-TC-spanner of a given directed graph. We completely resolve the approximability of 2-TC-spanners, showing that it is $\Theta(\log n)$ unless $\mathsf{P} = \mathsf{NP}$. For $k > 2$, we present a polynomial time algorithm that finds a k-TC-spanner with size within $O((n \log n)^{1-1/k})$ of the optimum. Our techniques also yield algorithms with the first nontrivial approximation ratio for well-studied problems on directed spanners when $k > 3$: DIRECTED k-SPANNER, CLIENT/SERVER DIRECTED k-SPANNER, and k-DIAMETER SPANNING SUBGRAPH. For constant $k \geq 3$, we show that the size of the sparsest k-TC-spanner is hard to approximate within a factor of $2^{\log^{1-\epsilon} n}$ for any $\epsilon \in (0, 1)$ unless $\mathsf{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{polylog}\, n})$. Finally, we study the size of the sparsest k-TC-spanners for H-minor-free graph families. Combining our constructions with our insight that 2-TC-spanners can be used for designing property testers, we obtain a monotonicity tester with $O(\log^2 n/\epsilon)$ queries for any poset whose transitive reduction, when viewed as an undirected graph, is free of a fixed minor. Previously, the best upper bound on the query complexity for such graphs was $O(\sqrt{n/\epsilon})$.

Key words. spanners, approximation algorithms, hardness of approximation, graph separators

AMS subject classifications. 68R10, 05C85, 68W25

DOI. 10.1137/110826655

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 Awerbuch [11] and Peleg and Schäffer [58] 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

*Received by the editors March 7, 2011; accepted for publication (in revised form) May 18, 2012; published electronically November 13, 2012. A preliminary version of this paper appeared in the Proceedings of SODA 2009 [18].
http://www.siam.org/journals/sicomp/41-6/82665.html

[†]Center for Computational Intractability, Princeton University, Princeton, NJ 08540 (arnabb@princeton.edu). Most of this research was done while this author was a graduate student at MIT and funded in part by NSF grants 0514771, 0732334, and 0728645 and by a DOE Computational Science Graduate Fellowship.

[‡]Department of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332 (elena@cc.gatech.edu). Most of this research was done while this author was a graduate student at MIT and funded in part by NSF grants CCR-0726525 and CCR-0829672.

[§]Department of Computer Science, KAIST, Daejeon, 30570, South Korea (kyomin@kaist.edu). This author's work was supported by the Engineering Research Center of Excellence Program of Korea Ministry of Education, Science, and Technology (MEST)/National Research Foundation of Korea (NRF) (grant 2010-0001713).

[¶]Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 (sofya@cse.psu.edu). This author's work was supported by National Science Foundation (NSF/CCF award 0729171 and NSF/CCF CAREER award 0845701).

[‖]IBM Almaden Research Center, San Jose, CA 94040 (dpwoodru@us.ibm.com).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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, 57, 73]. 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 [76]). Undirected spanners have numerous applications, such as efficient routing [26, 27, 60, 63, 72], simulating synchronized protocols in unsynchronized networks [59], parallel and distributed algorithms for approximating shortest paths [24, 25, 32], and algorithms for distance oracles [12, 73].

In the setting of directed graphs, two notions of spanners have been considered in the literature: the direct generalization of the above definition [58], and *roundtrip spanners* [27, 63]. 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. By diameter,[1] we mean the largest distance between a pair (u, v) of nodes in a directed graph such that v is reachable from u.

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 $d_H(u, v) \leq k$.*

The first property ensures that only the vertices that are connected in G will remain connected in the spanner. The second property guarantees that distances between connected pairs of vertices are small. The edges from the transitive closure of G that are added to G to obtain a TC-spanner are called *shortcuts*. Notice that a k-TC-spanner of G is a directed k-spanner of the transitive-closure of G. Nevertheless, TC-spanners are interesting in their own right due to the multiple TC-spanner-specific applications we present in section 1.3.

Our contributions. The contributions of this work fall into three categories: (1) We bring several diverse applications, including property testing, access control, and data structures, under the unifying framework of TC-spanners. These applications are discussed in section 1.3. (2) We initiate the study of the computational problem of finding the size of the sparsest k-TC-spanner of a given graph, which we refer to as k-TC-SPANNER. We present several approximability and hardness results for k-TC-SPANNER and its well studied variants. (3) Finally, we construct sparse TC-spanners for the family of H-minor free graphs, which include planar, bounded-treewidth, and bounded-genus graphs. Our constructions yield new algorithms and data structures for the application areas we discuss in section 1.3. Our results, described in items (2) and (3) above, are presented in section 1.2.

1.1. Related work.

For a directed graph G, we denote the size of (that is, the number of edges in) the sparsest k-TC-spanner of G by $S_k(G)$. To put the following results in proper context, observe that if G has n vertices, $S_k(G)$ is $O(n^2)$. This upper bound is tight, in general, as witnessed by the complete bipartite graph with $n/2$ vertices in each part, and all edges directed from one part to the other.

1.1.1. Approximability of directed spanner problems.

k-TC-SPANNER is a special case of a well-studied problem, called DIRECTED k-SPANNER, of finding the size of the sparsest k-spanner of a given (not necessarily transitively closed) directed graph. All algorithms for DIRECTED k-SPANNER immediately yield algorithms

[1]The definition of diameter used in this paper is nonstandard. The diameter is usually defined as the largest distance between a pair of nodes in a graph, and is set to infinity if a graph contains a pair of nodes with no path from one to the other.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

for k-TC-Spanner with the same approximation ratio. Elkin and Peleg [34] gave an $O(\log n)$-approximation algorithm for Directed 2-Spanner, and Kortsarz [50] showed that this approximation ratio cannot be improved unless $\mathsf{P} = \mathsf{NP}$. For $k = 3$, Elkin and Peleg [35] presented an $\tilde{O}(m^{1/3})$-approximation algorithm, where m is the number of edges in the graph. Thus, in the worst case, their approximation ratio is $\tilde{O}(n^{2/3})$. Their algorithm is quite complicated. For $k \geq 4$, approximation algorithms with ratios sublinear in n were known only in the undirected setting [58].

Elkin and Peleg [33] demonstrated that 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 $\mathsf{NP} \not\subseteq \mathrm{DTIME}(n^{poly \log n})$. Moreover, Elkin and Peleg [36] extended this result to $3 \leq k = O(n^{1-\delta})$ for all $\delta \in (0, 1)$. Thus, according to Arora and Lund's classification [48] of NP-hard problems, Directed k-Spanner is in class III, for $3 \leq k = O(n^{1-\delta})$. Moreover, [36] showed 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 longstanding open question in complexity theory, collapsing classes III and IV into a single class.

1.1.2. Approximability of more general problems. Dodis and Khanna [30] studied the problem of finding the minimum-cost subset of missing edges that can be added to a (directed) graph G, with costs and lengths associated to the missing edges, so as to ensure that that there is a path of length at most k between every pair of nodes (not only those connected in G). Observe that k-TC-Spanner is a special case of that problem: we can let G be the transitive reduction (see the definition in section 1.4) of the input graph to k-TC-Spanner, for all edges in the transitive closure of G set the length to 1 and cost to 1, and for the remaining edges set the length to k and cost to 0. Given this instance, the algorithm of Dodis and Khanna will produce a k-TC-spanner. However, their analysis guarantees only that the size of the resulting k-TC-spanner is at most $|E(G)| + O(OPT \cdot n \log k)$, where OPT is the number of missing edges that need to be added. If $|E(G)| = OPT = \Theta(n)$, their algorithm may return a k-TC-spanner with $\Omega(n^2)$ edges. Thus, in general, the resulting approximation ratio is no better than $O(n)$. Since their problem is more general, their hardness results do not apply to TC-spanners.

Chekuri et al. [23] gave an $O(p^{1/2+\epsilon})$-approximation algorithm for the directed Steiner network problem where, given a digraph and node pairs $(s_1, t_1), \ldots, (s_p, t_p)$, the goal is to connect all pairs with as few edges as possible. We can reduce k-TC-Spanner to this problem by specifying all comparable pairs of nodes in levels 1 and $k + 2$ in the $k + 1$-extension of G (see Definition 5.5 of [30]). However, their ratio is only $O(n^{1+\epsilon})$ when $p = \Omega(n^2)$, and thus the resulting ratio for k-TC-Spanner is no better than $O(n)$.

1.1.3. Structural results for $S_k(G)$. In order to reduce space used by parallel reachability oracles, Thorup [68] considered a special case of TC-spanners. He studied TC-spanners of a graph 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 [69], achieving stretch $O(\log^4 n)$ for this case. But the conjecture for general graphs was later disproved by Hesse [47], who showed that for all small $\epsilon > 0$, there exist graphs with $n^{1+\epsilon}$ edges for which all n^{ϵ}-TC-spanners must have $\Omega(n^{2-\epsilon})$ edges.

TC-spanners were also studied, implicitly in [21, 20, 5, 10, 22, 29, 75] and explicitly in [19, 70], for simple families of graphs, such as the directed line L_n, consisting of nodes $\{1, 2, \ldots, n\}$ and edges $\{(i, i + 1) : 1 \leq i \leq n - 1\}$, and arborescences (i.e., directed trees). Chandra, Fortune, and Lipton [21, 20] implicitly gave tight bounds

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

on $S_k(L_n)$ in the context of work on circuit complexity. They showed that the size of the sparsest k-TC-spanner for the directed line is $\Theta(n \cdot \lambda_k(n))$, where $\lambda_k(n)$ is the kth-row inverse Ackermann function (see section 1.4 for the definition). In particular, the sizes of the sparsest k-TC-spanners for the line for small k are $S_2(L_n) = \Theta(n \log n)$, $S_3(L_n) = \Theta(n \log\log n)$, and $S_4(L_n) = \Theta(n \log^* n)$. Alon and Schieber [5] also showed that the smallest k for which $S_k(L_n) = O(n)$ is $O(\alpha(n))$, where $\alpha()$ is the inverse Ackermann function defined in section 1.4. Note that $S_k(L_n) \geq n-1$ for all k, since all edges of the form $(i, i+1)$ have to be present in a TC-spanner to ensure the same connectivity as in L_n. Alon and Schieber [5], Chazelle [22], and Thorup [70] showed that S_k for arborescences is asymptotically the same as for the line.

There has also been study of graph-theoretic notions that are similar in spirit to, although distinct from, TC-spanners. For example, Cohen [25] constructed sparse *hop sets*. A (d, ϵ)-hop set for a weighted undirected graph $G = (V, E)$ is a weighted graph $G' = (V, E \cup E')$ such that for any two vertices $u, v \in V$, the graph G' contains a path from u to v of at most d edges and of weight within $1 + \epsilon$ of the minimum weight path in G. Hop sets also reduce diameter by adding extra edges, but the problems are incomparable because TC-spanners are defined on *unweighted* and *directed* graphs.

1.2. Our results. In this section, we describe our algorithms and inapproximability results for k-TC-Spanner, Directed k-Spanner and well-studied variants of these problems, and our construction of sparse TC-spanners for the family of H-minor free graphs. Tables 1.1 and 1.2 summarize our (and previously known) results on the approximability of k-TC-Spanner. Subsequent algorithmic developments are described in section 1.5. All our hardness and structural results were still the best known when this paper was last revised.

1.2.1. 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 linear programming and sampling, and gives an $O((n \log n)^{1-1/k})$-ratio for k-TC-Spanner. In fact, the algorithm also

Table 1.1
Summary of algorithmic results on k-TC-Spanner.

Setting of k	Approximability	Notes	Reference
$k = 2$	$O(\log n)$		[34]
$k = 3$	$O(n^{2/3} \operatorname{polylog} n)$		[33]
$k > 2$	$O((n \log n)^{1-1/k})$	Applies to Directed k-Spanner	This work
$k = \Omega\left(\frac{\log n}{\log\log n}\right)$	$O\left(\frac{n \log n}{k^2 + k \log n}\right)$	Separation from Directed k-Spanner	This work

Table 1.2
Summary of hardness results on k-TC-Spanner, all from this paper.

Setting of k	Inapproximability	Assumption	Notes
$k = 2$	$\Omega(\log n)$	$\mathsf{P} \neq \mathsf{NP}$	Matches the upper bound
constant k	$\Omega(2^{\log^{1-\epsilon} n})$	$\mathsf{NP} \subseteq \mathrm{DTIME}(n^{poly \log n})$	Improvement implies breakthrough
$k = n^{1-\epsilon}$ for all $\epsilon > 0$	$\Omega(1+\gamma)$ for some $\gamma = \Omega(1/k)$	$\mathsf{P} \neq \mathsf{NP}$	

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

applies to the more general Directed k-Spanner problem. This resolves the open question of finding a sublinear approximation ratio for the Directed k-Spanner problem for $k > 3$, described as a "challenging direction" by Elkin and Peleg [35]. Moreover, our algorithm for $k = 3$ is simpler than the $O(n^{2/3} \operatorname{polylog} n)$-approximation algorithm of [35].

Our algorithm produces the spanner by taking a union of edge sets of two graphs. The first graph is obtained by formulating the problem as an integer program, solving a linear programming relaxation, and rounding the solution. The second graph is formed by sampling vertices from G uniformly at random and growing breadth-first search (BFS) arborescences around them. We give a simple greedy algorithm that derandomizes the sampling procedure. The sampling step, consisting of growing BFS arborescences rooted at random vertices, without the combination with linear programming, was previously employed by Aingworth et al. [4] and Dor, Halperin, and Zwick [31] in efficient algorithms for approximating all pairs shortest paths in unweighted undirected graphs. Their algorithms construct a weighted graph that emulates distances in a given unweighted undirected graph with small additive error. The combination of linear programming and sampling was used by Kortsarz and Peleg [52] to build low-degree undirected 2-spanners. Their sampling consists of adding each edge to the spanner independently and their linear program is a special case of the linear program we use (with a different objective).

Our algorithmic technique also yields sublinear approximation ratios for well-studied variants of the Directed k-Spanner problem: $O((n \log n)^{1-1/k})$ for k-Diameter Spanning Subgraph and $O(n^{1-1/(2k-1)+\epsilon})$ for Client/Server Directed k-Spanner (see [34] for definitions). For the last problem, we use some additional ideas that were formulated by Feldman, Kortsarz, and Nutov in [40] in the course of designing an algorithm for the Directed Steiner Forest problem.

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 Elkin and Peleg [36, Theorem 6.6] showed that Directed $\sqrt{n}$-Spanner is $2^{\log^{1-\epsilon} n}$-inapproximable. Moreover, Hesse [47] 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.

1.2.2. Inapproximability of k-TC-Spanner. We present three results on the hardness of k-TC-Spanner, applicable for different values of k. First, we prove for $k = 2$ that the $O(\log n)$ ratio of Kortsarz and Peleg [51] is optimal unless $\mathsf{P} = \mathsf{NP}$. Next, for constant $k > 2$, we show that k-TC-Spanner is inapproximable within a factor of $2^{\log^{1-\epsilon} n}$, for all $\epsilon \in (0, 1)$, unless $\mathsf{NP} \subseteq \mathrm{DTIME}(n^{\operatorname{polylog} n})$. 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 by Elkin and Peleg [36], collapse classes III and IV in Arora and Lund's classification. Finally, we show that for any $k < n^{1-\epsilon}$ for any $\epsilon > 0$, k-TC-Spanner is inapproximable to within a factor of $1 + \gamma$ for some $\gamma = \Omega(1/k)$, unless $\mathsf{P} = \mathsf{NP}$.

Our $2^{\log^{1-\epsilon} n}$-hardness matches the known hardness for Directed k-Spanner for constant $k > 2$. 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 better than $\Omega(1)$-hardness for k-TC-Spanner. Intuitively, our lower bound is much harder to prove

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

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. We use the generalized butterfly graphs, defined by Woodruff [74], and the broom graphs to construct hard instances of k-TC-SPANNER. Our reduction transforms the MIN-REP problem to 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 a generalized butterfly are well-structured, which allows us to analyze many different routes possible in the transitive closure.

1.2.3. Structural results. Finally, we construct sparse TC-spanners for a specific graph family: $\mathcal{H}$-minor-free graphs. A graph H is a *minor* of G if H is a (not necessarily induced) subgraph of a graph obtained from G by a sequence of edge contractions. A graph family $\mathcal{F}$ is *minor-closed* if it contains every minor of every graph in $\mathcal{F}$. For a finitely large graph set $\mathcal{H}$, a minor-closed family $\mathcal{F}$ is $\mathcal{H}$*-minor-free* if for all $H \in \mathcal{H}$, $H \notin \mathcal{F}$. For example, planar graphs are $\mathcal{H}$-minor-free for $\mathcal{H} = \{K_5, K_{3,3}\}$. Note that this means planar graphs are, in particular, K_5-minor-free. In addition to planar graphs, $\mathcal{H}$-minor-free families include bounded-treewidth and bounded-genus graphs, explicitly studied in applications described in section 1.3. Indeed, the celebrated Robertson–Seymour theorem shows that *any* minor-closed family of graphs can be defined by a finite set of forbidden minors. All $\mathcal{H}$-minor-free graphs are sparse, i.e., have a linear number of edges [55].

We show that if the transitive reduction of a directed graph G, disregarding edge orientations, is an undirected graph that excludes a fixed minor H, then there is an efficient construction of a 2-TC-spanner of G of size $O(n \log^2 n)$ and, more generally, for any constant k, a k-TC-spanner of size $O(n \cdot \log n \cdot \lambda_k(n))$, where $\lambda_k(\cdot)$ is the kth-row inverse Ackermann function. The former result is tight because it gives a $O(n \log^2 n)$ bound on the size of a 2-TC-spanner of any planar graph and, in particular, a directed 2-dimensional grid, for which Berman et al. [13] subsequently proved a matching lower bound. Our result on TC-spanners of H-minor-free graphs allowed us to drastically improve monotonicity testers of Fischer et al. [42]. The application to monotonicity testing and our improvement are described in section 1.3.

The main idea in our construction is the use of the *path separators* for undirected H-minor free graphs due to Abraham and Gavoille [1]. However, although the separators are undirected 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, described in section 1.1.3. 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.

As mentioned in section 1.1.3, Thorup [69] constructed polylog(n)-TC-spanners of size $O(n)$ for digraphs with planar transitive reductions. His idea was to apply a TC-spanner construction for arborescences in a certain recursive way. There are two reasons why Thorup's method does not easily extend to give our results. First, his reduction from planar graphs to arborescences depends on the Jordan curve theorem, and so, it cannot be modified straightforwardly to work for H-minor-free graphs. Second, his reduction blows up the stretch of the spanner by a polylogarithmic factor, while our construction produces spanners of constant stretch.

1.3. Applications of TC-spanners. We describe three applications that use sparse TC-spanners: monotonicity testing, key management in an access hierarchy, and data structures for computing partial products in a semigroup.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

1.3.1. Monotonicity testing. Monotonicity of functions [3, 29, 38, 41, 42, 44, 46] is one of the most studied properties in property testing [45, 64]. Fischer et al. [42] prove that testing monotonicity is equivalent to several other testing problems. Let V_n be a poset of n elements and let $G_n = (V_n, E)$ 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 at least an ϵ 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, proposed by Dodis et al. [29], is based on the sparsest 2-TC-spanner for that graph. Implicit in the proof of Proposition 9 in [29] 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 1.2. *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 $O(\frac{s(n)}{\epsilon n})$.*

Proof. The tester selects $\frac{8s(n)}{\epsilon n}$ edges of the 2-TC-spanner H uniformly at random. It queries the input 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_n by $TC(G_n)$. 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 $TC(G_n)$. This implies (see [42], 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 $TC(G_n)$ [29]. 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 two 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 two edges in M can be mapped to one violated edge in H. Thus, the 2-TC-spanner H has $\geq \frac{\epsilon n}{4}$ violated edges, as required. ∎

The fact that H is a 2-TC-spanner is crucial for the proof. If it was a k-TC-spanner for $k > 2$, the path of length k from x to y might not have any violated edges incident to x or y, even if $f(x) > f(y)$. Consider $G_{2n} = (V_{2n}, E)$, where $V_{2n} = \{x_1, \ldots, x_{2n}\}, E = \{(x_i, x_n) \mid i < n\} \cup (x_n, x_{n+1}) \cup \{(x_{n+1}, x_j) \mid j > n + 1\}$. G_{2n} is a 3-TC-spanner of itself. Now set $f(x_i) = 1$ for $i \leq n$ and $f(x_i) = 0$ otherwise. Clearly, this function is $\frac{1}{2}$-far from monotone, but only one edge, (x_n, x_{n+1}), is violated in the 3-TC-spanner.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

By Lemma 1.2, 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 one due to Fischer et al. [42]. Indeed, we achieve testers with $O(\log^2 n/\epsilon)$ queries, whereas the best previously known bound was $O(\sqrt{n/\epsilon})$.

1.3.2. Key management in an access hierarchy. In the problem of key management in an access hierarchy, 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 [10]). One approach to the key management problem [9, 10, 65] 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, [10] 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 [10] are TC-spanners for arborescences 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.

1.3.3. Partial semigroup products, Euclidean spanners, and other applications. Yao [75] and Alon and Schieber [5] studied space-efficient data structures for the following problem: Preprocess elements $\{s_1, \dots, s_n\}$ 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 $1 \le i < j \le n$ with at most k queries to a small database of precomputed partial products. This problem reduces to finding a sparsest k-TC-spanner for a directed line L_{n+1}.

Chazelle [22] and Alon and Schieber [5] 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 us to compute the product of elements associated with all vertices on the path from i to j, for all vertices i, j in T. As before, only k queries to the data structure are allowed for each product computation. The generalized problem reduces to finding a sparsest k-TC-spanner for an arborescence T' obtained from T by appending a new vertex to each leaf, and then selecting an arbitrary root and directing all edges away from it. A k-TC-spanner for T' with $s(n)$ edges yields a preprocessing scheme with space complexity $s(n)$ for computing products on T with at most $2k$ queries as follows. The database stores a product $s_{v_1} \circ \cdots \circ s_{v_t}$ for each k-TC-spanner edge (v_1, v_{t+1}) if the endpoints of that edge are connected by the path $v_1, \dots, v_t, v_{t+1}$ in T'. Let $LCA(u, v)$ denote the lowest common ancestor of u and v in T. To compute the product corresponding to a path from u to v in T, we consider 2 cases: (1) if u is an ancestor of v (or vice versa) in T, query the products corresponding to the k-TC-spanner edges on the shortest path from u to a child of v (from v to a child of u, respectively); (2) otherwise, make queries corresponding to the k-TC-spanner edges on the shortest path from $LCA(u, v)$ to a child of u and on the shortest path from a child of $LCA(u, v)$ nearest to u to a child of u. This gives a total of at most $2k$ queries.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TC-spanners for arborescences have also been a key technical tool for constructing low-diameter Euclidean spanners, an extensively studied object in computational geometry [7, 56, 67, 66]. Here, one is given an undirected weighted complete graph on a vertex set S, identified with a set of points in $\mathbb{R}^d$, where $d \geq 1$. The weight of each edge is the Euclidean distance between the endpoints. A *Euclidean t-spanner of diameter k* is a subset E of edges which, for any two points $x, y \in S$, contains a path from x to y of total weight at most $t \cdot \|x - y\|$ and with at most k edges. It is easy to verify that a k-TC-spanner for the directed line L_n gives a Euclidean 1-spanner of diameter k for n points in 1 dimension. Extending this idea, Arya et al. [7] showed that Euclidean spanners for point sets in higher dimensions can be constructed from 1-*spanners of diameter k for tree metrics.* A 1-spanner of diameter k of a weighted undirected tree $T = (V, E)$ is a graph $G = (V, E \cup E')$, where E' is a set of weighted undirected edges over the vertex set V, such that for any two vertices $u, v \in V$, the unweighted distance between u and v in G is at most k and the weighted distance between u and v in G equals the weighted distance in T. A 1-spanner of diameter k for a tree is clearly also a k-TC-spanner for the arborescence obtained by directing all tree edges outwards from the root, and an argument analogous to the one in the previous paragraph shows that a k-TC-spanner of an arborescence gives a 1-spanner of diameter $2k$ for the underlying undirected tree. Through this connection, the previously described TC-spanner constructions in [5] allowed Arya et al. [7] to get 1-spanners for tree metrics and, from there, low-diameter Euclidean spanners for arbitrary dimensions.

Another beautiful application of TC-spanners was discovered by Thorup in [68], although he realized only later the connection to the previous work on the semigroup problem. Thorup showed that if there exists a k-TC-spanner of size s for a directed graph G on n vertices, then there is a concurrent-read-concurrent-write (CRCW) parallel algorithm with running time $O(k)$, number of processors s, and space $O(n)$ that decides reachability from a distinguished vertex v in G. In separate work, Bodlaender, Tel, and Santoro [19] studied the *nonreversing diameter* problem, which is intimately related to the Euclidean spanner problem mentioned above and, hence, also to the TC-spanner problem. There, they discuss some other algorithmic applications which benefit from sparse TC-spanner constructions.

1.4. Preliminaries and notation.

We write $u \rightsquigarrow_G v$ to denote that a vertex v is reachable from a vertex u in a graph G. When the graph is clear from the context, we omit G. The *transitive closure* of a directed 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 $(v, u) \in TC(G)$.

A *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, Garey, and Ullman [2], $TR(G)$ can be computed efficiently via a greedy algorithm.[2] For directed acyclic graphs $TR(G)$ is unique. We say G is *transitively reduced* if $TR(G) = G$.

In the context of building TC-spanners, we call an edge a *shortcut edge* if it is in $TC(G)$ but not in G.

A digraph G is *weakly connected* if replacing each directed edge in G with an undirected edge results in a connected undirected graph. A digraph is *strongly connected*

[2]The algorithm contracts each strongly connected component C to a vertex $v(C)$ to get a supergraph H, obtains a supergraph H' by greedily removing arcs in H that do not change its transitive closure, and finally uncontracts the $v(C)$, choosing a representative vertex of C to be incident to the edges incident to $v(C)$ in H'.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

if each vertex in the graph is reachable from every other vertex via a directed path. The graph of strongly connected components of a digraph G is the digraph obtained by contracting each strongly connected component into one vertex, while maintaining all the edges between these components.

We will need a variant of the family of Ackermann functions. Define the following two families of functions:

$$\begin{aligned} A(0,j) &= 2j && \text{for } j \geq 1,\\ A(i,0) &= 1 && \text{for } i \geq 1,\\ A(i,j) &= A(i-1, A(i,j-1)) && \text{for } i,j \geq 1 \end{aligned}$$

$$\begin{aligned} B(0,j) &= j^2 && \text{for } j \geq 1,\\ B(i,0) &= 2 && \text{for } i \geq 1,\\ B(i,j) &= B(i-1, B(i,j-1)) && \text{for } i,j \geq 1. \end{aligned}$$

The *inverse Ackermann function* is $\alpha(n) = \min\{i : A(i,i) \geq n\}$. For $i = 2k$, the *ith-row inverse* is $\lambda_i(n) = \min\{j : A(k,j) \geq n\}$ and for $i = 2k+1$, $\lambda_i(n) = \min\{j : B(k,j) \geq n\}$. Explicitly, for all $n \geq 0$, $\lambda_0(n) = \lceil n/2 \rceil$, $\lambda_1(n) = \lceil \sqrt{n} \rceil$, $\lambda_2(n) = \lceil \log n \rceil$, $\lambda_3(n) = \lceil \log\log n \rceil$, $\lambda_4(n) = \lceil \log^* n \rceil$, $\lambda_5(n) = \lfloor \frac{1}{2} \log^* n \rfloor$, etc. One useful identity satisfied by these functions is that $\lambda_k(n) = 1 + \lambda_k(\lambda_{k-2}(n))$.

1.5. New developments. After the publication of the conference version of this paper in SODA 2009 [18], several follow-up works appeared. Our two algorithms were improved, new structural results and more applications of such results were presented, and a new variant of k-TC-Spanner was studied.

First, our approximation ratio of $\tilde{O}(n^{1-1/k})$ for Directed k-Spanner for $k > 2$ was improved to $\tilde{O}(n^{1-1/\lceil k/2 \rceil})$ by Berman, Raskhodnikova, and Ruan [16]. Then Dinitz and Krauthgamer [28] gave an alternative $\tilde{O}(n^{1/2})$-approximation algorithm for $k = 3$ and achieved $\tilde{O}(n^{2/3})$ ratio for $k > 3$. Later, Berman et al. [14] improved the approximation ratio to $\tilde{O}(n^{1/3})$ for $k = 3$ and to $\tilde{O}(\sqrt{n})$ for $k > 3$. The algorithm for $k = 3$ in [14] also gave the first improvement in the approximation ratio for Undirected 3-Spanner in two decades. We note that all algorithms in [28, 14] build off of our hybrid approach of solving a linear program and randomly sampling vertices and growing BFS arborescences around the samples. The new ideas in these subsequent works can also be used to improve the approximation ratio for the Client/Server Directed k-Spanner problem considered in this paper. In addition, our $O((n \log n)/(k^2 + k \log n))$ approximation ratio for k-TC-Spanner was improved to $O(n/k^2)$ in [16].

Second, Bhattacharyya et al. [17] and Jha and Raskhodnikova [49] proposed new applications of structural results on TC-spanners to local monotonicity reconstruction and to testing and local reconstruction of the Lipschitz property, respectively. Bhattacharyya et al. [17] also presented nearly tight bounds on the size of TC-spanners of the hypercube and the hypergrid. Their lower bound for the hypergrid was improved by Berman et al. [13]. In addition, Berman, Raskhodnikova, and Ruan [16] gave structural results relating the sizes of the sparsest 2-TC-spanners and k-TC-spanners for $k > 2$.

Third, Atallah et al. [8], De Santis, Ferrara, and Masucci [65], and Berman et al. [13] studied a new variant of TC-spanners, called *Steiner TC-spanners*, where new (Steiner) nodes can be inserted into a TC-spanner, in addition to new shortcut edges, while ensuring that every pair of non-Steiner nodes is connected by a path in the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TC-spanner iff it was connected in the original graph. They showed that for many digraphs, Steiner nodes can drastically reduce the size of a k-TC-Spanner [8, 65, 13], though for hypergrids their effect is limited [13].

For a more detailed description of recent results, we refer the reader to the survey by Raskhodnikova [61].

1.6. Organization. Section 2 describes our approximation algorithms for k-TC-Spanner and related problems. In section 3, we present our hardness results. Section 4 describes our construction for H-minor-free graphs.

2. Algorithms for k-TC-Spanner and related problems.

In this section, we present two approximation algorithms for k-TC-Spanner. The first one, presented in section 2.1, gives a better approximation ratio for small k and also applies to other related problems. The second one, presented in section 2.2, gives a better approximation for large k, and is specific to k-TC-Spanner.

For both algorithms, we assume that the input digraph G is *weakly connected*. If this does not hold, our algorithms can be run on each weakly connected component separately.

2.1. Algorithm for Directed k-Spanner. Our $O((n \log n)^{1-1/k})$-approximation for Directed k-Spanner for arbitrary k is based on a new combination of linear programming and sampling. Our technique also achieves an $O((n \log n)^{1-1/k})$ ratio for the k-Diameter Spanning Subgraph and an $\tilde{O}(n^{1-1/(2k-1)+\epsilon})$ ratio for the Client/Server Directed k-Spanner, problems considered in [35]. Previously, no $o(n)$-approximation algorithms were known for these problems.

Our result for Directed k-Spanner is stated below in Theorem 2.1. To achieve the same result for k-TC-Spanner, it suffices to run the algorithm on the transitive closure of the input digraph. The extensions to the other problems are described in this section in Theorem 2.8.

Theorem 2.1. *For any (not necessarily constant) $k \geq 2$, there is a deterministic polynomial time algorithm for* Directed k-Spanner *with approximation ratio $O((n \log n)^{1-1/k})$.*

Our algorithm produces the spanner by taking a union of edge sets of two graphs. The first graph is obtained by formulating the problem as an integer program, solving a linear programming (LP) relaxation, and rounding the solution. The second graph is formed by sampling vertices from G uniformly at random and growing BFS arborescences around them; later, we show how to derandomize the sampling procedure.

We start by describing the LP relaxation. Let G be the input digraph. One can introduce binary edge variables x_e for each edge e in G, and binary path variables y_P for each path P of length at most k in G. One can impose 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. Then add constraints $\sum_P y_P \geq 1$ for all edges $(u, v) \in G$, where the sum is over paths P of length at most 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 seems to be quite bad, which may be why an LP approach had not been considered before. Indeed, at an intuitive level, 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 $\Theta(1/n)$. The second problem with this approach is that the number of variables and the size of the constraints grow exponentially with k. We resolve both problems by treating edges (u, v) with a large number of paths of length at most k from u to v separately from the remaining edges.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

DEFINITION 2.2 (central edges). *Let $b = (n \log n)^{1-1/k}$, where n is the number of nodes in the input graph G and k is the desired stretch of a TC-spanner of G. We call an edge (u, v) of G* central *if there are more than b (not necessarily simple) paths from u to v in G containing at most k edges. Otherwise, (u, v) is* noncentral.

We (1) introduce constraints $\sum_P y_P \geq 1$ only for noncentral edges (u, v); (2) sample additional nodes uniformly at random and grow BFS arborescences around them to ensure that central edges (u, v) are "covered" by short paths in the spanner. Since there are numerous paths of length at most k from u to v, they contain many vertices, and we are likely to sample one of them.

DEFINITION 2.3 ($BFS(w)$). *Let w be a vertex in G. Then $BFS_{out}(w)$ is a shortest path out-arborescence rooted at w, consisting of edges in G directed away from w. Similarly, $BFS_{in}(w)$ is a shortest path in-arborescence rooted at w, consisting of edges in G directed towards w. $BFS(w)$ is the set of edges obtained as the union of $BFS_{out}(w)$ and $BFS_{in}(w)$. It contains at most $2(n - 1)$ edges.*

If w is on a path of length at most k from u to v, then the path from u to v along the edges in $BFS(w)$ also has length at most k. Therefore, combining the solutions resulting from the LP rounding and from sampling ensures that both central and noncentral edges (u, v) are "covered" by short paths in the spanner.

Proof of Theorem 2.1. Consider the following integer programming (IP) formulation. For each edge e in the input digraph G, we introduce a variable x_e indicating whether x_e occurs in the k-spanner. For each noncentral edge (u, v) and each such path P, we introduce a variable y_P indicating whether all of the edges of P occur in the k-spanner.

$$\begin{array}{llll} & \text{minimize} & \sum_{e \in G} x_e & \\ (2.1) & \text{subject to} & \sum_{P \text{ from } u \text{ to } v,\ |P| \leq k} y_P \geq 1 & \forall \text{ noncentral } (u,v) \in G \\ & & y_P \leq x_e & \forall P \text{ and } \forall e \in P \\ & & x_e,\ y_P \in \{0,1\} & \forall e\ \forall P. \end{array}$$

The first constraint ensures that the spanner contains at least one path of length at most k spanning each noncentral edge (u, v), while the second constraint allows a path to be included only if each of its edges is also in the spanner. Thus, any directed k-spanner satisfies the constraints of this program.

This integer program can be computed in polynomial time, and therefore, is of polynomial size. More specifically, for each edge (u, v), one can compute b shortest paths in $O(m + n \log n + b \log b)$ time, where n is the number of nodes and m is the number of edges in G [37], and hence, the time needed to write down the program is $O(m^2)$.

```
ALGORITHM. Spanner Generation (Input: directed graph $G = (V, E)$ on $n$ vertices, stretch $k$).
    1. Take integer program (2.1), relax the constraints $x_e, y_P \in \{0,1\}$ to $x_e \in [0,1], y_P \in \mathbb{R}$, and let $x^*, y^*$ be the solution to the resulting LP.
    2. Initialize the spanner edge set $E_H$ to $\emptyset$, and let $b = (n \log n)^{1-1/k}$.
    3. For each edge $e$ in $E$, if $x^*_e \geq 1/b$, add $e$ to $E_H$.
    4. Sample a set $Z = \{z_1, \ldots, z_r\}$ of $r = 3 \ln 2 \cdot b$ vertices from $V$ uniformly with replacement.
    5. For each $i \leq r$, add $BFS(z_i)$ to $E_H$.
    6. Output $H = (V, E_H)$.
```

LEMMA 2.4. *With probability at least $1 - 1/n$, H is a directed k-spanner of G.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. Consider a noncentral edge (u, v) in G. By the first constraint of (2.1), there exists a path P from u to v of length $\leq k$ for which $y^*_P \geq 1/b$. Then, by the second constraint of (2.1), $x^*_e \geq 1/b$ for all edges e on path P. Thus, path P is included in H in step 3 of Spanner Generation.

Now consider a central edge (u, v) in G. Let $W_{u,v}$ be the set of vertices lying on (not necessarily simple) paths of length at most k from u to v. Let $\mathcal{W}$ be the collection of sets $W_{u,v}$ for all central edges (u, v) in G. Recall that a *hitting set* for a collection of sets is a set that intersects with all sets in the collection.

Claim 2.5. *With probability at least* $1 - 1/n$, *set* Z *chosen in step* 4 *of* Spanner Generation *is a hitting set for* $\mathcal{W}$.

Proof. Fix a central edge (u, v), and let $s = |W_{u,v}|$. The number of $u - v$ paths of length at most k that can be formed from s vertices is at most s^{k-1}. So, $s^{k-1} \geq (n \log n)^{1-1/k}$, and therefore, $s \geq (n \log n)^{1/k}$. The probability that $Z \cap W_{u,v} = \emptyset$ is at most

$$(1 - s/n)^r \leq e^{-rs/n} = e^{-3 \ln 2 \log n} = 1/n^3.$$

By a union bound, with probability at least $1 - 1/n$, all central edges (u, v) in G satisfy $Z \cap W_{u,v} \neq \emptyset$. ▯

If Z is a hitting set for $\mathcal{W}$, then for each central edge (u, v), let $z(u, v)$ be an arbitrary element in $Z \cap W_{u,v}$. Then the path from u to v via $z(u, v)$ along the edges of $BFS(z(u, v))$ is of length at most k. Indeed, since $z(u, v) \in W_{u,v}$, there is a path P of length at most k from u to v which contains $z(u, v)$. The path from u to v along the edges of $BFS(z(u, v))$ cannot be longer than P. Therefore, H is a directed k-spanner of G whenever the event in Claim 2.5 occurs. ▯

Lemma 2.6. *Let* OPT *be the size of an optimal directed* k*-spanner of* G. *The number of edges in* H *is*

$$O((n \log n)^{1-1/k} OPT).$$

Proof. Let OPT' be the optimum of the LP relaxation of (2.1) used in the Spanner Generation algorithm. Clearly, $OPT' \leq OPT$, since the LP is a relaxation. In step 3 of Spanner Generation, at most $b \cdot OPT' \leq b \cdot OPT$ edges are added to H. In step 5, $O(rn) = O(b \cdot n)$ edges are added to H. So, $|H| = O(b \cdot (OPT + n))$. Recall that we assumed without loss of generality that G is weakly connected. With this assumption, $OPT \geq n - 1$. Therefore, $|H| = O((n \log n)^{1-1/k} OPT)$. ▯

Lemmas 2.4 and 2.6 show that Spanner Generation is a randomized $O((n \log n)^{1-1/k})$-approximation algorithm for Directed k-Spanner. The algorithm runs in polynomial time, as the linear program is of polynomial size in the number of variables and constraints.

Derandomization. Spanner Generation can be derandomized by greedily choosing vertices in step 4, instead of sampling them at random. First, for each central edge (u, v) in G, construct the set $W_{u,v}$ of all vertices lying on (not necessarily simple) paths of length at most k from u to v. This can be done by computing $BFS(w)$ for each vertex w in G, and checking if the path from u to v via w along the edges of $BFS(w)$ has length at most k.

The hitting set Z for collection $\mathcal{W}$ can be constructed deterministically using a standard claim, given as Lemma 2.7. The number of sets in $\mathcal{W}$ is equal to the number of central edges, which is $O(n^2)$. As we argued in the proof of Claim 2.5, each set in

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$\mathcal{W}$ contains at least $(n \log n)^{1/k}$ vertices. Therefore, the resulting hitting set Z has size

$$O\left(\frac{n \log n^2}{(n \log n)^{1/k}}\right) = O\left((n \log n)^{1-1/k}\right) = O(b),$$

the same as in the randomized algorithm.

LEMMA 2.7. *Given a collection of m sets, each containing at least s elements, coming from a universe of size n, one can efficiently construct a hitting set Z of size $O((n \log m)/s)$.*

Proof. Pick elements in Z greedily, and after selecting each element, delete sets in which this element appears from the collection. Namely, always pick an element that appears in the largest number of the remaining sets.

Since each set contains at least s elements, by averaging, there is an element which occurs in at least an $\frac{s}{n}$ fraction of the remaining sets. Hence, after t steps, we delete at least a $(1-s/n)^t \leq e^{-st/n}$ fraction of the sets. Setting t to the smallest integer exceeding $\frac{n \ln m}{s}$ makes this fraction less than $\frac{1}{m}$. Then all sets are deleted after t steps, ensuring that all sets in the original collection intersect Z. ∎

This completes the proof of Theorem 2.1. ∎

Extension to variants of DIRECTED *k*-SPANNER. Our algorithm can also be extended to solve variants of DIRECTED k-SPANNER. In the k-DIAMETER SPANNING SUBGRAPH, given a directed graph G, one has to find a sparse subgraph of G (not $TC(G)$) such that all pairs of vertices (u, v) for which v is reachable from u are connected by a path of length at most k. In the ALL-CLIENT DIRECTED k-SPANNER problem, for a given directed graph G and for a given subset of edges called the *server edges*, one has to find a subgraph of G consisting of only server edges that is a k-spanner of G. Finally, in the CLIENT/SERVER DIRECTED k-SPANNER problem, one has both a set of *client edges* and a set of server edges, and the goal is to construct a subgraph consisting of server edges that spans each client edge with stretch at most k.

THEOREM 2.8. *For all constant $k > 2$, there are deterministic $O((n \log n)^{1-1/k})$-approximation algorithms for k-TC-SPANNER, k-DIAMETER SPANNING SUBGRAPH, and ALL-CLIENT DIRECTED k-SPANNER.*

Proof. As we discussed, k-TC-SPANNER can be solved by running an algorithm for DIRECTED k-SPANNER on the transitive closure of the input graph.

For the k-DIAMETER SPANNING SUBGRAPH problem, we consider *central* pairs of comparable vertices rather than just edges, defined as in Definition 2.2. The rest of the algorithm remains unchanged.

For the ALL-CLIENT DIRECTED k-SPANNER problem, we modify our algorithm by changing the definition of a *central* edge (Definition 2.2): we call an edge (u, v) of G *central* if it is a client edge and there are more than b (not necessarily simple) paths of server edges from u to v in G containing at most k edges. (Remaining client edges are called *noncentral*.) We introduce variables x_e only for server edges e in G and consider BFS arborescences with respect to the graph of the server edges. Notice that the size of an optimal solution for this problem is still at least n. The rest of the algorithm and the analysis remains unchanged. ∎

The above approach does not directly lead to an algorithm for the CLIENT/SERVER DIRECTED k-SPANNER problem, essentially because we no longer have $OPT \geq n-1$ if the number of client edges is small, and hence, we cannot add arborescences to handle central edges as this might be too costly. Below, we modify our algorithm, using some ideas from [40], to solve the CLIENT/SERVER DIRECTED k-SPANNER problem.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 2.9. *For all constant $k > 2$ and for all fixed $\epsilon > 0$, there is a randomized polynomial time $O(n^{1-1/(2k-1)+\epsilon})$-approximation algorithm for* CLIENT/SERVER DIRECTED k-SPANNER.

Proof. Given an input digraph $G = (V, E)$ on n vertices, we can assume without loss of generality that we know τ such that $OPT \leq \tau < 2 \cdot OPT$. This is so, because we can repeat the entire argument below for every value of $\tau \in \{1, 2, 4, \ldots, 2^{\lceil \log_2 n^2 \rceil}\}$ and stop at the smallest value of τ for which we obtain the desired spanner. This incurs only a logarithmic loss in the time complexity, which can be ignored.

DEFINITION 2.10. *A* server path *is a path in G that consists entirely of server edges.*

Let $b = n^{1-1/(2k-1)}$. We change the definition of central edges (Definition 2.2) as follows: a client edge (u, v) is *central* if there are more than b server paths from u to v of length at most $\min(k, \tau/b)$. For a central edge (u, v), let $W_{u,v}$ be the set of vertices lying on server paths of length at most $\min(k, \tau/b)$ from u to v, and let $\mathcal{W}$ be the collection of sets $W_{u,v}$ for all central edges (u, v). Since the total number of server paths of length at most $\min(k, \tau/b)$ from u to v is at most $|W_{u,v}|^{k-1}$, we have $|W_{u,v}|^{k-1} \geq b$ or, equivalently, $|W_{u,v}| \geq b^{1/(k-1)}$. Now, the same argument as in Claim 2.5 shows that a set Z of $(3n \ln n)/b^{1/(k-1)}$ random vertices from V is a hitting set for $\mathcal{W}$ with probability at least $1 - 1/n$, since for any fixed client edge (u, v), the probability that $Z \cap W_{u,v} = \emptyset$ is at most

$$(1 - |W_{u,v}|/n)^{3n \ln n/b^{1/(k-1)}} \leq e^{-(3 \ln n)|W_{u,v}|/b^{1/(k-1)}} \leq 1/n^3.$$

For each randomly sampled vertex $z \in Z$ and each vertex v in G incident to a client edge, add the edges of a shortest server path from z to v if it is of length at most $\min(k, \tau/b)$, and also, the edges of a shortest server path from v to z if it is of length at most $\min(k, \tau/b)$. The total number of edges added in this step is at most $(3n \ln n)/b^{1/(k-1)} \cdot n \cdot \tau/b = \tilde{O}(b) \cdot \tau = \tilde{O}(b) \cdot OPT$, using the fact that the number of vertices incident to client edges is at most n. Let H' be the set of edges added. With probability at least $1 - 1/n$, for every central edge (u, v), H' contains a path from u to v of length at most k.

Let F be the set of noncentral client edges of G. We now need to find server paths of length at most k spanning each edge in F. For a set of edges H'', let $\mathsf{density}(H'')$ denote the ratio of $|H''|$ to the number of edges in F for which H'' contains a server path of length at most k. (Note that the density can be a number greater than 1.) We show below how to find H'' such that $\mathsf{density}(H'')$ is at most $O(bn^\epsilon) \cdot OPT/|F|$. If such H'' can be found, we can continue in a greedy fashion by removing from F the set of edges for which there are already server paths of length at most k in H'' and repeating the procedure. A well-known covering argument then shows that this greedy procedure eventually results in a set H of edges that contains a server path of length at most k from u to v for *every* noncentral client edge (u, v), and the size of H is $O(bn^\epsilon \log n) \cdot OPT$. The union of H and H' then forms the resulting spanner, proving the theorem. (The $\log n$ factor in the approximation ratio can be removed by making ϵ slightly larger.)

Now, we describe how to find H''. For the purposes of analysis, fix an optimal solution O with $OPT \in [\tau, 2\tau)$ edges, and let L be the set of client edges (u, v) for which every server path of length at most k from u to v in O is of length more than τ/b. Note that L may be empty (for instance, if $k < \tau/b$). There are two cases: (i) $|L| < |F|/2$, and (ii) $|L| \geq |F|/2$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Case (i) $|L| < |F|/2$. In this case, $|F - L| \geq |F|/2$. This means that for at least $|F|/2$ edges $(u, v) \in F$, there exists a server path from u to v of length at most $\min(k, \tau/b)$. In this case, we use an LP relaxation similar to that in the proof of Theorem 2.1. For an edge $(u, v) \in F$, let $\mathcal{P}_{u,v}$ denote the set of server paths from u to v of length at most $\min(k, \tau/b)$. For each server edge e in G, we have a variable x_e indicating whether e occurs in the spanner. For each edge $(u, v) \in F$, we have a variable $f_{(u,v)}$ indicating whether there is a server path of length at most $\min(k, \tau/b)$ from u to v. Finally, for each server path $P \in \mathcal{P}_{u,v}$ for any $(u, v) \in F$, there is a variable y_P indicating whether all the edges of P occur in the spanner. Now, consider the following IP:

$$\begin{array}{lll}
\text{minimize} & \sum\limits_{\text{server edge } e} x_e & \\
\text{subject to} & \sum_{P \in \mathcal{P}_{u,v}} y_P \geq f_{(u,v)} & \forall (u, v) \in F \\
& y_P \leq x_e & \forall P \in \bigcup_{(u,v)\in F} \mathcal{P}_{(u,v)} \text{ and } \forall e \in P \\
& \sum_{(u,v)\in F} f_{(u,v)} \geq |F|/2 & \\
& x_e,\ y_P, f_{(u,v)} \in \{0, 1\} & \forall e\ \forall P\ \forall (u, v) \in F.
\end{array}$$

It is clear that O satisfies this IP. Also, $|\mathcal{P}_{u,v}| \leq b$ since (u, v) is noncentral, and so, the IP can be computed in polynomial time [37]. Now, we relax the IP to an LP by relaxing the constraints $x_e,\ y_P, f_{(u,v)} \in \{0, 1\}$ to $x_e, y_P, f_{(u,v)} \in [0, 1]$. Let x^*, y^*, f^* be a solution to this LP. We now make a simple observation (analogous to Lemma 3.6 in [40]).

CLAIM 2.11. $|\{(u, v) \in F \mid f^*_{(u,v)} \geq 1/4\}| \geq |F|/3$.

Proof. Suppose not. Then

$$\sum_{(u,v)\in F} f^*_{(u,v)} < \frac{|F|}{3} \cdot 1 + \frac{2|F|}{3} \cdot \frac{1}{4} = \frac{|F|}{2},$$

which would violate the third LP constraint. ∎

Thus, by the first LP constraint, for at least $|F|/3$ edges $(u, v) \in F$, we have $\sum_{P \in \mathcal{P}_{u,v}} y^*_P \geq 1/4$. So, for every such edge (u, v), since (u, v) is noncentral, by averaging, there exists a server path $P \in \mathcal{P}_{u,v}$ such that $y^*_P \geq 1/(4b)$. By the second LP constraint, $x^*_e \geq 1/(4b)$ for every server edge e on any such path P. Hence, if we include in H'' any server edge e with $x^*_e \geq \frac{1}{4b}$,

$$\mathsf{density}(H'') \leq (4b\tau)/(|F|/3) = 12b\tau/|F|.$$

Case (ii) $|L| \geq |F|/2$. In this case, for at least $|F|/2$ client edges (u, v), every server path in O from u to v is of length more than τ/b. For each $(u, v) \in F$, let $P_{u,v}$ be a path of length at most k in O from u to v. So, $\sum_{(u,v)\in F} |P_{u,v}| \geq \tau|F|/(2b)$. Since the total number of server edges in O is less than 2τ, it must be the case that some edge e occurs in at least $|F|/(4b)$ such paths $P_{u,v}$, since otherwise, $\sum_{(u,v)\in F} |P_{u,v}| < 2\tau \cdot |F|/(4b) = \tau|F|/(2b)$. This leads to the following claim.

CLAIM 2.12. *An edge set in a directed graph is called a* junction tree *if it is the union of an in-arborescence and an out-arborescence (not necessarily edge disjoint), both rooted at the same vertex. When $|L| \geq |F|/2$, there exists a junction tree H'' such that $\mathsf{density}(H'') \leq 8b\tau/|F|$.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. As argued above, there must exist an edge e such that for $|F|/(4b)$ distinct pairs $(u,v) \in F$, there is a server path from u to v of length at most k which contains e. Let r be one of the endpoints of e. Now, note that without loss of generality, the $|F|/(4b)$ server paths can be assumed to form a junction tree H'' rooted at r. If two outgoing paths from r intersect at a vertex v, only one of the paths from r to v can be included in H'' without changing the number of pairs $(u,v) \in F$ for which H'' contains a server path of length at most k. The same argument holds for incoming paths. Hence, H'' is a junction tree of density at most $2\tau/(|F|/(4b)) = 8b\tau/|F|$. □

At this stage, we can use the following result of [23].

LEMMA 2.13 (Lemma 3.3 in [23]). *For any fixed $\epsilon > 0$, there is a polynomial-time algorithm that, given a graph $G = (V,E)$ on n vertices, constructs a junction tree $J \subseteq E$ satisfying $\mathsf{density}(J) = O(n^\epsilon) \cdot \mathsf{density}(J^*)$, where J^* is a minimum density junction tree for G.*

If $|L| \geq |F|/2$, Claim 2.12 shows that there is a junction tree of density $O(b\tau/|F|)$. So, the algorithm from Lemma 2.13 is guaranteed to produce a junction tree of density $O(n^\epsilon b\tau/|F|)$.

Therefore, no matter which case holds, we can find an edge set of density

$$O(n^\epsilon b\tau/|F|) = O(bn^\epsilon) \cdot OPT/|F|$$

by running the algorithms for both cases and letting H'' be the edge set which has the smaller density. As argued before, this proves the theorem. □

2.2. Algorithm for k-TC-SPANNER for large k.

For large k, we present a better approximation algorithm, which is specific to the k-TC-SPANNER problem.

THEOREM 2.14. *For any k, there exists a deterministic polynomial time algorithm for k-TC-SPANNER with approximation ratio*

$$O\left(\frac{n \log n}{k^2 + k \log n}\right).$$

Proof. Let G be the input digraph. Assume without loss of generality that G is weakly connected. Then $S_k(G) \geq n-1$, and to prove the theorem it is enough to construct a k-TC-spanner H of G of size

$$|TR(G)| + O\left(\frac{n^2 \log n}{k^2 + k \log n}\right). \tag{2.2}$$

Since all graphs with the same transitive closure give rise to the same k-TC-SPANNER problem, we can assume without loss of generality that G is transitively reduced.

Let $k_1 = \left\lfloor \frac{k}{2} \right\rfloor$ and $k_2 = k - k_1 - 1$. To construct H, we first select a subset Z of vertices, which we call *leaders*, such that every two comparable vertices at distance k_1 in G are connected by a shortest path via a leader. For each leader $z \in Z$, we select a subset V_z of vertices reachable from z, which we call *followers*. Our k-TC-spanner H contains all edges in G and shortcut edges connecting each leader to its followers. The followers of z are selected in such a way that for every vertex v reachable from z, there is a follower z' of z that connects to v via a path of length at most k_2 in G.

To see that the resulting H is a k-TC-spanner, consider a vertex pair (u,v) in G with $u \rightsquigarrow_G v$. If u is connected to v via a path of length at most k in G, then the same holds in H. Otherwise, consider an arbitrary path $(u = v_0, v_1, \ldots, v = v_\ell)$ in G, as shown in Figure 2.1. By definition of the leader set Z, there is a leader z on a shortest path from $u = v_0$ to v_{k_1}. That is, the distance from u to z is at most k_1,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

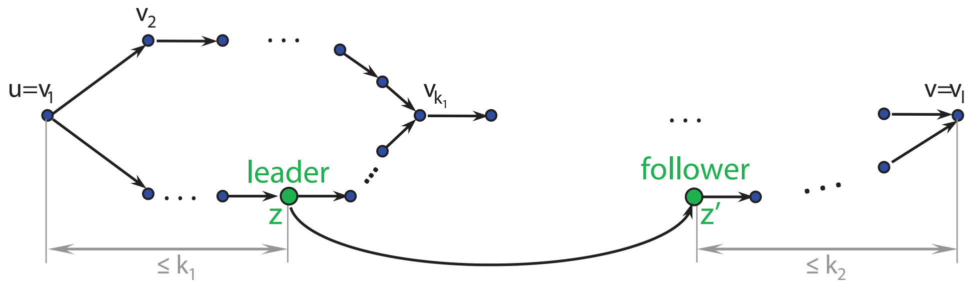

FIG. 2.1. *A path of length at most $k = k_1 + k_2 + 1$ between u and v in the proof of Theorem* 2.14.

and $u \rightsquigarrow z \rightsquigarrow v_{k_1} \rightsquigarrow v$. Since v is reachable from z, there is a follower z' of z that connects to v via a path of length at most k_2. Since H contains an edge from z to its follower z', it must contain a path of length at most $k_1 + 1 + k_2 = k$ from u to v via (z, z').

It remains to show how to construct sufficiently small sets of leaders and followers. First, we construct the set Z of leaders. Consider a pair of comparable vertices (u, v) at distance k_1 in G. Fix a shortest path from u to v, and let $W_{u,v}$ be the set of vertices on that path. Let $\mathcal{W}$ be the collection of sets $W_{u,v}$ for all comparable u, v at distance k_1 in G. We construct Z to be the hitting set of $\mathcal{W}$, using Lemma 2.7. Then indeed some shortest path from u to v contains a vertex in Z for all u, v at distance k_1 in G. The number of leaders is $|Z| = O(\frac{n \log n}{k})$ because each $W_{u,v}$ contains $k_1 + 1 = \Omega(k)$ vertices, the universe size, i.e., the total number of vertices, is n and the number of sets in $\mathcal{W}$ is at most the number of vertex pairs, that is, $O(n^2)$. Observe that the size of Z cannot exceed n. Therefore,

$$|Z| = O\left(\min\left(n, \frac{n \log n}{k}\right)\right) = O\left(\frac{n \cdot \frac{n \log n}{k}}{\max\left(n, \frac{n \log n}{k}\right)}\right)$$
$$= O\left(\frac{\frac{n \log n}{k}}{1 + \frac{\log n}{k}}\right) = O\left(\frac{n \log n}{k + \log n}\right).$$

To construct the set V_z of followers of vertex z, first compute the out-arborescence $BFS_{out}(z)$ described in Definition 2.3. For each $i \in \{0, 1, \ldots, k_2 - 1\}$, let V_z^i denote the set of vertices at distance $i \pmod{k_2}$ from z. Since the number of vertices in $BFS_{out}(z)$ is at most n, one of the sets V_z^i has at most n/k_2 vertices. We let V_z be the smallest set V_z^i. By construction, for every vertex v reachable from z, there is a vertex $z' \in V_z$ that connects to v via a path of length at most k_2 in G.

Since $|V_z| \le \frac{n}{k_2}$ for all z, the size of the constructed k-TC-spanner is

$$|G| + O\left(\frac{n^2 \log n}{k^2 + k \log n}\right).$$

Since G is transitively reduced, this is equal to the expression in (2.2). ▯

3. Hardness results for *k*-TC-SPANNER.

This section presents our hardness results for k-TC-SPANNER. In section 3.1, we prove $\Omega(\log n)$ hardness of approximation for 2-TC-SPANNER. Section 3.2 describes our main technical contribution, the

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

hardness result for constant $k \geq 3$. In section 3.3, we give a general NP-hardness result for large k, for which our hardness of approximation results do not apply.

We start by describing why previous inaproximability results for the Directed k-Spanner problem do not imply similar inaproximability results for the k-TC-Spanner problem. 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 [50, 33, 36] 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 from [50] 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 [47] 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 [74] as the main building blocks of our reductions.

Definition 3.1 (generalized butterfly graphs). *A* generalized butterfly *of diameter k and width n is a digraph whose vertices are identified with coordinates $[n^{1/k}]^k \times [k+1]$, and whose edges are between vertices $u = (u_1, \ldots, u_k, i)$ and $v = (v_1, \ldots, v_k, i+1)$ iff for all $j \neq i$, $u_j = v_j$. We say a vertex $(u_1, \ldots, u_k, i)$ is in strip i. Therefore, vertices in strips $i \in [k]$ have out-degrees equal to $n^{1/k}$, and vertices in*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

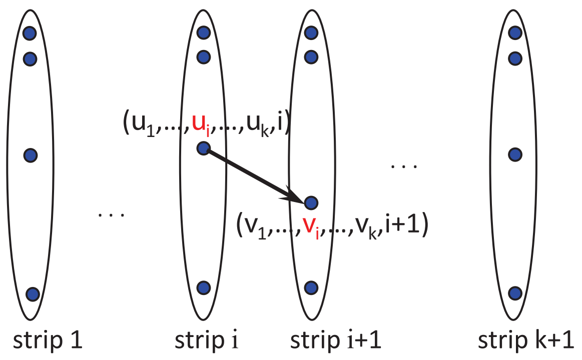

FIG. 3.1. *A generalized butterfly.*

strips $j \in \{2, \ldots, k+1\}$ have in-degrees equal to $n^{1/k}$. (See Figure 3.1 *for an illustration.)*

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, as we rigorously argue later, 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, at least $n^{1+2/k}$ shortcuts are needed to reduce the diameter to $k-1$.

3.1. $\Omega(\log n)$-hardness of 2-TC-Spanner.

The goal of this section is to prove the following theorem.

THEOREM 3.2. *For all k, such that $2 \leq k = o(\frac{\lg n}{\lg \lg n})$, it is NP-hard to approximate the size of the sparsest k-TC-spanner of a given digraph G within a ratio of $O(\frac{1}{k} \log n)$. In particular,* 2-TC-SPANNER *is $\Omega(\log n)$-inapproximable.*

Our proof uses a reduction from a variant of SET COVER, called (a, b, c)-NICE SET COVER. Before defining this problem, we define other variants of SET COVER. An instance of (a, b)-SET COVER, consists of a bipartite graph $G = A \cup B$, with $|A| = a$ and $|B| = b$. An instance of a-BALANCED SET COVER consists of a bipartite graph $G = A \cup B$, with $|A| = |B| = a$. An instance of (a, c)-BALANCED BOUNDED SET COVER consists of a bipartite graph $G = A \cup B$, with $|A| = |B| = a$ and such that the degrees of the vertices in A are at most c. Finally, an instance of (a, b, c)-NICE SET COVER consists of a bipartite graph $G = A \cup B$, with $|A| = a, |B| = b$. B can be partitioned into disjoint sets B_i such that $B = \cup_{i=1}^{a} B_i$, $|B_i| = \frac{b}{a}$, assuming $\frac{b}{a}$ is an integer. G must satisfy the property that if $v \in A$ is adjacent to $w \in B_i$, for some $1 \leq i \leq a$, then v is adjacent to every element of B_i. Moreover, v is adjacent to at most c sets B_i. A solution to all these SET COVER variants is a minimum number of vertices in A that cover all the vertices in B.

LEMMA 3.3. *It is NP-hard to approximate a solution to (n^a, n^b, n^c)-*NICE SET COVER *to within a ratio of $\gamma c \cdot \log n$ for some constant γ, where $0 < c \leq a \leq b$.*

Proof. We will need the following fact, proved in [62]. Earlier, this result was shown under the weaker assumption that NP $\not\subseteq$ DTIME$(n^{O(\log \log n)})$ [39, 54].

FACT 3.4. *There is a constant $d > 0$ for which it is NP-hard to approximate a solution to (n^d, n)-*SET COVER *to within a ratio of $\gamma \log n$, for some $\gamma > 0$.*

CLAIM 3.5. *It is NP-hard to approximate a solution to n-*BALANCED SET COVER *to within a ratio of $\gamma \log n$, for the same γ as above.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Proof. By Fact 3.4, (n^d, n)-Set Cover is not approximable within a factor of $\gamma \log n$, unless $\mathsf{P} = \mathsf{NP}$. Using a reduction from (n^d, n)-Set Cover, if $|A| = n^d < n$, transform this instance into an instance where $|A| = |B|$ by padding A with dummy vertices. If $|A| > n$, transform this instance into an instance where $|A| = |B|$ by padding the set B with dummy vertices and connecting them to all vertices in A. ∎

Claim 3.5 allows us to now prove hardness of Balanced Bounded Set Cover.

Claim 3.6. *It is* NP*-hard to approximate a solution to* (n^a, n^c)-Balanced Bounded Set Cover *to within a ratio of* $\gamma c \cdot \log n$, *where* γ *is from Claim* 3.4 *above.*

Proof. The proof is similar to that of Lemma 2.3 of [50]. Take an instance I of n^c-Balanced Set Cover, which is NP-hard to approximate within a factor of $\gamma c \log n$ by Claim 3.5. Replicate this instance n^{a-c} times to create an instance I' of (n^a, n^c)-Balanced Bounded Set Cover. Clearly, the solution to I' must consist of n^{a-c} copies of the solution to the underlying I. Hence, I' is also inapproximable to within a ratio of $\gamma c \log n$. ∎

To complete the proof of the lemma, notice that a set M is a solution to an instance of (n^a, n^b, n^c)-Nice Set Cover iff M is a solution to the instance of (n^a, n^c)-Balanced Bounded Set Cover, resulted from compressing each set B_i into a single vertex b_i, $1 \le i \le n^a$. By Claim 3.6 above, it follows that (n^a, n^b, n^c)-Nice Set Cover is not approximable within a ratio of $\gamma c \log n$, unless $\mathsf{P} = \mathsf{NP}$. ∎

We now prove the main theorem of this section.

Proof of Theorem 3.2. Let $\alpha = 1 + \frac{3}{2k}$ and $\beta = \frac{1}{5k}$. Given $G_1 = V_{k+1} \cup V_{k+2}$, an instance of (n, n^α, n^β)-Nice Set Cover, transform it into the following $(k+2)$-partite graph $G = V_1 \cup V_2 \cup \cdots \cup V_{k+1} \cup V_{k+2}$, with edges directed from V_i to V_{i+1}. Let $|V_i| = n$, $1 \le i \le k+1$, and recall that by definition $V_{k+2} = n^\alpha$. The induced subgraph on $V_1 \cup V_2 \cup \ldots V_{k+1}$ is the butterfly graph of diameter k and width n. Then $|G| \le kn^{1+\frac{1}{k}} + n^{\alpha+\beta} = O(n^{1+\frac{17}{10k}})$ edges, because $k = o(\frac{\lg n}{\lg \lg n})$. Notice that there are indeed at most $n^{\alpha+\beta}$ edges from V_{k+1} to V_{k+2} since there are n vertices in V_{k+1}, each of degree at most $n^{\beta+\alpha-1}$.

Let OPT_S denote the size of the minimal k-TC-spanner for G, and let OPT_{NSC} denote the size of the solution to the Nice Set Cover instance G_1.

Lemma 3.7. $OPT_S = \Theta(OPT_{NSC}\ n^{\frac{2}{k}})$.

Proof. First, we show that there is a k-TC-spanner H of G such that $|H| = \Theta(OPT_{NSC}\ n^{\frac{2}{k}})$ edges. Then we show that any k-TC-spanner of G must have $\Omega(OPT_{NSC}\ n^{\frac{2}{k}})$ edges.

Notice that the only pairs of vertices of G that are not already at distance at most k are the comparable vertices u, v, with $u \in V_1$ and $v \in V_{k+2}$. In order to connect such pairs by a directed path of length at most k, we need "shortcut" edges between different levels V_i and V_j, $i + 2 \le j$. Without loss of generality, we may assume that the only shortcut edges used are those connecting vertices in V_i to V_{i+2}, for some i's. Indeed, a shortcut edge connecting a vertex $u \in V_i$ to a vertex $v \in V_j$, where $j > i+2$ can be replaced with one edge connecting $u \in V_i$ to a vertex $w \in V_{i+2}$ that is an ancestor of v. In this way, all paths from V_1 to V_{k+2} that previously had a path of length at most k still have a path of length at most k. Define an edge $e = (u, v)$ to be a *type* i *edge* if $u \in V_i$ and $v \in V_{i+2}$. We will say that a vertex u *reaches* an edge $e = (v, w)$ if there is a path from u to v. We next build a k-TC-spanner of G with $\Theta(OPT_{NSC}\ n^{\frac{2}{k}})$ edges. Let H be the smallest k-TC-spanner of G which uses only shortcut edges of type $k-1$.

Claim 3.8. $|H| = \Theta(n^{\frac{2}{k}}\ OPT_{NSC})$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. Let O be a set of vertices in V_{k+1} that is an optimal solution to the (n, n^α, n^β)-Nice Set Cover instance. Connect each vertex $v \in O$ to the set A_v of all the $n^{2/k}$ ancestors of v from level V_{k-1}. Direct these edges from A_v to v. Notice that we added $OPT_{NSC}\ n^{\frac{2}{k}}$ edges, and the new graph H' is a k-TC-spanner. Indeed, each vertex $u \in V_1$ is comparable to each vertex $v \in O$, and thus, there is a vertex $w \in A_v$ that is comparable to u. This implies that for every $u \in V_1$, there is a path of length $k-1$ to each of the vertices of O, resulting in a path of length k to each vertex in V_{k+2}.

To show that H (the minimum size k-TC-spanner with shortcuts only of type $k-1$) needs at least $OPT_{NSC}\ n^{\frac{2}{k}}$ edges on top of those in G, assume otherwise. For $v \in V_{k-1}$, let $n(v)$ be the number of type $k-1$ edges leaving from v. By assumption, $\sum_{v \in V_{k-1}} n(v) < OPT_{NSC}\ n^{\frac{2}{k}}$. Each vertex in $v \in V_{k-1}$ has exactly $a(v) = n^{1-\frac{2}{k}}$ ancestors in V_1. For $u \in V_1$, let $e(u)$ be the total number of type $k-1$ shortcuts leaving from its descendants in V_{k-1}. Since there exists a path of length k from u to each vertex in V_{k+2}, it follows that $e(u) \geq OPT_{NCS}$. Notice that $\sum_{v \in V_{k-1}} n(v)a(v) = \sum_{u \in V_1} e(u) \geq OPT_{NCS}\ n$. This implies that $\sum_{v \in V_{k-1}} n(v) \geq OPT_{NCS}\ n^{\frac{2}{k}}$, a contradiction to our assumption. We conclude that $|H| = |G| + OPT_{NSC}\ n^{\frac{2}{k}}$. Next we show that $|H| = \Theta(n^{\frac{2}{k}}\ OPT_{NSC})$. Indeed, OPT_{NSC} is, by construction, the same as the size of the optimal solution to an (n, n^β)-Balanced Bounded Set Cover instance, where we must cover n vertices on the right with n vertices of degree at most n^β on the left. This implies that $OPT_{NSC} \geq n^{1-\beta} = n^{1-\frac{1}{5k}}$. Now, $|G| = \Theta\ (n^{1+\frac{17}{10k}})$ and $|H| = |G| + OPT_{NSC}\ n^{\frac{2}{k}}$. Since $OPT_{NSC} n^{\frac{2}{k}} \geq n^{\frac{2}{k}+1-\frac{1}{5k}} = n^{1+\frac{18}{10k}}$, this implies that $|H| = \Theta(OPT_{NSC}\ n^{\frac{2}{k}})$. ∎

Let M be a sparsest spanner of G which possibly uses shortcut edges of types other than $k-1$. Assume for the sake of contradiction that $|M| < \frac{1}{4}\ n^{\frac{2}{k}}\ OPT_{NSC}$. A vertex $u \in V_1$ can reach $v \in V_{k+2}$ in at most k steps by using shortcut edges either of type $i \leq k-1$ or of type k. We will show that, under our assumption, there are many vertices in V_1 that can reach at most $\frac{1}{2}\ OPT_{NSC}$ vertices in V_{k+1} by using edges only of some types $i < k$. Moreover, there are many vertices in V_1 that reach only $n^{\frac{1}{k}} OPT_{NSC}$ edges of type k. That will be enough to argue that a contradiction must occur, allowing us to conclude that $|M| = \Theta(n^{\frac{2}{k}}\ OPT_{NSC})$.

Claim 3.9. *Let R be the set of vertices in V_1 that can reach less than $\frac{1}{2}\ OPT_{NSC}$ vertices $v \in V_{k+1}$ in at most $k-1$ steps in M. Then $|R| > \frac{n}{2}$.*

Proof. For each vertex $u \in V_1$ and $v \in V_{k+1}$, define an indicator variable $X_{u,v}$ which is 1 iff there is a shortcut edge along the unique path from u to v in G. Consider a type i shortcut edge $e = (v_i, v_{i+2})$, with $v_i \in V_i$ and $v_{i+2} \in V_{i+2}$. Then there are $n^{\frac{i-1}{k}}$ vertices u in V_1 such that there is a path from u to v_1. Moreover, there are $n^{\frac{k-i-1}{k}}$ vertices $v \in V_{k+1}$ such that there is a path from v_{i+2} to v. Thus, this shortcut edge e can set at most $n^{\frac{i-1}{k}+\frac{k-i-1}{k}} = n^{1-\frac{2}{k}}$ different $X_{u,v}$ to 1. By assumption, there are less than $\frac{1}{4}\ n^{\frac{2}{k}} OPT_{NSC}$ shortcut edges of types i, where $i \leq k-1$. It follows that less than $\frac{1}{4}\ n\ OPT_{NSC}$ different $X_{u,v}$'s can be set to 1. For $u \in V_1$, let $n(u)$ be the number of vertices $v \in V_{k+1}$ that u can reach in less than k steps. Thus, $\mathbb{E}_{u \in V_1}\ [n(u)] < \frac{1}{4}\ OPT_{NSC}$. By Markov's inequality, $Pr_{u \in V_1}[n(u) \geq \frac{1}{2}\ OPT_{NSC}] < \frac{1}{2}$. This implies that more than $\frac{1}{2}$ of the vertices $u \in V_1$ can reach less than $\frac{n}{2}\ OPT_{NSC}$ vertices $v \in V_{k+1}$ in less than k steps. Therefore, $|R| > \frac{n}{2}$. ∎

For $u \in V_1$, let $t(u)$ be the number of type k edges that u reaches in M.

Claim 3.10. *Let S be the set of vertices $u \in V_1$ such that $t(u) < \frac{1}{2}\ n^{\frac{1}{k}}\ OPT_{NSC}$. Then $|S| > \frac{n}{2}$.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Proof. Assuming $|M| < \frac{1}{4}\, n^{\frac{2}{k}}\ OPT_{NSC}$, there are at most $\frac{1}{4}\, n^{\frac{2}{k}}\ OPT_{NSC}$ edges of type k. Each $v \in V_k$ has exactly $n^{1-\frac{1}{k}}$ ancestors in V_1, and therefore $\sum_{u\in V_1} t(u) < n^{1-\frac{1}{k}}\ \frac{1}{4}\ n^{\frac{2}{k}}\ OPT_{NSC} = \frac{1}{4} n^{1+\frac{1}{k}} OPT_{NSC}$. Thus, $\mathbb{E}_{u\in V_1}[\ t(u)] < \frac{1}{4} n^{\frac{1}{k}} OPT_{NSC}$ and by Markov's inequality, $Pr_{u\in V_1}[t(u) < \frac{1}{2} n^{\frac{1}{k}} OPT_{NTS}] > \frac{1}{2}$. □

Let $T = R \cap S$. The two claims above imply $|T| \geq 1$. Now we argue that a vertex $v \in T$ cannot reach some vertices in V_{k+2}. Recall that an instance of $(n, n^{\alpha}, n^{\beta})$-NICE SET COVER is associated with an underlying instance of (n, n^{β})- BALANCED BOUNDED SET COVER, by compressing each set B_i on the right into a single vertex. As we already observed in the proof of Lemma 3.3, a set of elements on the left is a solution to the NICE SET COVER problem iff it is also a solution to the associated BALANCED BOUNDED SET COVER problem. Suppose we remove $\frac{1}{2}\ n^{\frac{1}{k}} OPT_{NSC}$ vertices from V_{k+2}. This corresponds to removing at most $\frac{1}{2} n^{\frac{1}{k}+1-\alpha} OPT_{NSC} = \frac{1}{2}\ n^{-\frac{1}{2k}} OPT_{NSC} = o(1)\ OPT_{NSC}$ vertices from the universe of the related (n, n^{β})-BALANCED BOUNDED SET COVER instance. Let OPT_{BSC} be the size of a solution to this new SET COVER problem. Then $OPT_{BSC} \geq (1 - o(1))\ OPT_{NSC}$.

Suppose then that $v \in T$ could cover all of the elements in V_{k+2}. Each such vertex $v \in T$ can cover vertices in V_{k+2} in exactly two ways: (1) from the $\frac{1}{2} OPT_{NSC}$ vertices it reaches in V_{k+1} via paths of length $< k$ using type $i < k$ edges, and (2) by at most $\frac{1}{2} n^{\frac{1}{k}} OPT_{NSC}$ type k edges it can reach. Thus, $OPT_{BSC} \leq \frac{1}{2}\ OPT_{NSC}$, which is a contradiction since $OPT_{BSC} \geq (1 - o(1))\ OPT_{NSC}$. Thus, $v \in T$ cannot reach all of V_{k+2}, and so the optimal k-TC-spanner on G must have size at least $n^{\frac{2}{k}} OPT_{NSC}/4$. We can them conclude that $|M| = \Theta(n^{\frac{2}{k}}\ OPT_{NSC})$. □

Suppose now that we could approximate the size of the sparsest k-TC-spanner within $\gamma_1 \log n$ for some $\gamma_1 > 0$. Then, since $|M| = \Theta(n^{\frac{2}{k}} OPT_{NSC})$, we could approximate a solution to $(n, n^{\alpha}, n^{\beta})$-NICE SET COVER within $\gamma_2 \log n$, for some $\gamma_2 > 0$. By Lemma 3.3, $(n, n^{\alpha}, n^{\beta})$-NICE SET COVER cannot be approximated within $\gamma\beta \log n = O(1/k) \log n$, unless $\mathsf{P} = \mathsf{NP}$. Therefore, the size of the sparsest k-TC-spanner cannot be approximated within a factor $(\gamma_3 \log n)/k$, for some $\gamma_3 > 0$, unless $\mathsf{P} = \mathsf{NP}$. □

3.2. $2^{\log^{1-\epsilon} n}$-hardness for constant $k \geq 3$.

The following is the formal statement of our inapproximability result for $k \geq 3$.

THEOREM 3.11. *For all fixed $\epsilon \in (0, 1)$ and constant $k \geq 3$, the size of the sparsest k-TC-spanner cannot be approximated to within a factor of $2^{\log^{1-\epsilon} n}$ unless $\mathsf{NP} \subseteq DTIME(n^{\mathrm{polylog}\, n})$.*

Reduction from MIN-REP. To get $2^{\log^{1-\epsilon} n}$-inapproximability, we reduce from the MIN-REP problem, defined next.

DEFINITION 3.12 (the MIN-REP problem). *An (n, r, d, m)-*MIN-REP *instance is a bipartite graph satisfying the following properties:*

1. *It has maximum degree d.*
2. *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 $i \in [r]$ (see Figure 3.2).*
3. *To describe the last parameter m, call a vertex* isolated *if its degree is 0, and* nonisolated *otherwise. Let $m(\mathcal{A}_i)$ be the inverse of the fraction of nonisolated vertices in $\mathcal{A}_i$. Then m is the minimum such $m(\mathcal{A}_i)$.*

The supergraph *of this graph is defined as the graph with (super)nodes $\mathcal{A}_1, \ldots, \mathcal{A}_r$, $\mathcal{B}_1, \ldots, \mathcal{B}_r$, and* (super)edges *$(\mathcal{A}_i, \mathcal{B}_j)$ present iff there is a node in $\mathcal{A}_i$ adjacent to a node in $\mathcal{B}_j$.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

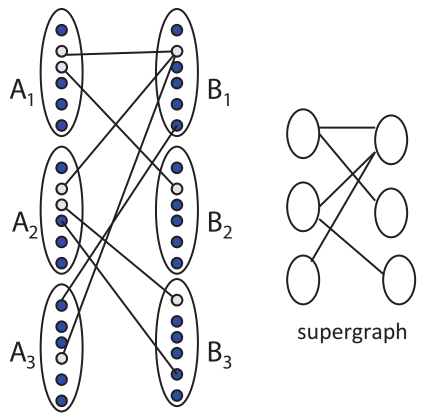

FIG. 3.2. *A* MIN-REP *instance and the corresponding supergraph.*

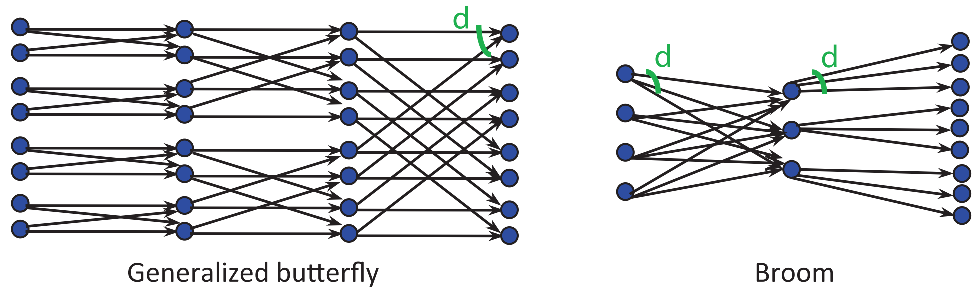

FIG. 3.3. *Graphs used in the reduction from* MIN-REP *to* k-TC-SPANNER.

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.*

Our proof relies on the following results of [50].

FACT 3.13 (hardness of approximating MIN-REP [50]). *For all $\epsilon \in (0, 1)$, there is no polynomial time algorithm for the* MIN-REP *problem with approximation ratio $2^{\log^{1-\epsilon} n}$ unless $\mathsf{NP} \subseteq DTIME(n^{\text{polylog}\, n})$.*

A first attempt. 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 xth *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. See Figure 3.3. 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). See Figure 3.4.

We would like to argue that the sparsest 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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

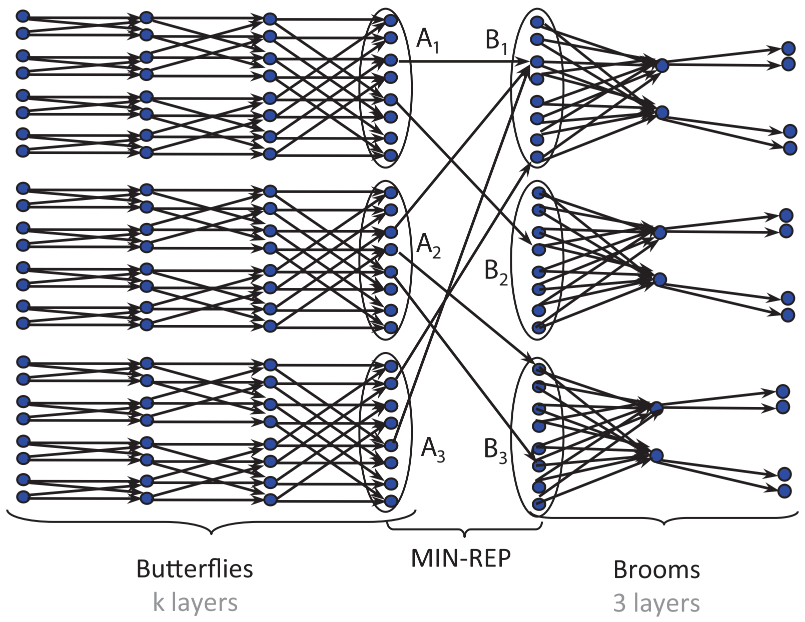

FIG. 3.4. *Reduction from* MIN-REP.

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 that 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.

Improving the first attempt. 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. Since 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 the optimal solution and the four parameters of the MIN-REP instance each lie in a narrow range. In Theorem 3.14, 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 applying 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. (See Theorem 3.14.)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

In analyzing our lower bound on the size of a k-TC-spanner for the graph $\mathcal{G}$ the isolated vertices play a central role, as they help control the number of pairs from the first and last layers of $\mathcal{G}$ that use alternative paths (as defined above). More precisely, computing the number of pairs from these layers that use specific shortcuts amounts to computing products of the out-degrees of vertices in various layers of the graph. We will ensure that the butterflies are connected to the Min-Rep instance in such a way that the 1-shadow vertices are connected to a large fraction of isolated neighbors, this essentially eliminating many potential paths. This feature allows us argue that the number of pairs that use alternative paths is small compared to OPT and can thus be ignored (see Lemma 3.18 for a precise statement). An equivalent way of achieving this feature would have been to decrease the out-degrees of the vertices in the 1-shadows. For the sake of elegance of presentation, however, we preferred to rather use full butterfly graphs combined with isolated vertices.

Recap of the overall strategy. We give a final overview of the overall strategy and then proceed to prove the individual steps. First, we show a variant of Min-Rep, which we call noise-resilient Min-Rep, is NP-hard. This gives us careful control over the parameters r, d, and m. We further specialize the instance of noise-resilient Min-Rep to what we call specialized Min-Rep, which further divides the clusters in noise-resilient Min-Rep into identical groups. This will help bound the overall number of paths connecting comparable vertices of certain types. Next, we show how to build an instance of the k-TC-spanner problem from specialized Min-Rep, which involves attaching generalized butterflies to the left set of vertices of the specialized Min-Rep instance, attaching broom graphs to the right set of vertices of the specialized Min-Rep instance, and directing all edges from left to right.

At this point we have a directed graph $\mathcal{G}$. We can easily show that there exists a k-TC-spanner of $\mathcal{G}$ of a certain size related to the optimum of the specialized Min-Rep instance; this is done in our rep-cover spanner lemma. The harder part is in Lemma 3.17, where we show that this k-TC-spanner is almost optimal in its number of edges. To do so we classify all paths between comparable vertices depending on the types of shortcut edges that their near-shortest paths contain. We first remove superedges in the specialized Min-Rep instance which have many comparable vertices that have paths that go through them and use shortcut edges of a type that we do not want. We show by our choice of parameters in the specialized Min-Rep instance that this results in only removing $o(OPT)$ superedges; see our path analysis lemma (here we use the isolated vertices in the third case analysis to bound the number of comparable vertices which use shortcut edges of a certain type). Next, after removing this small number of superedges, we show how to take a k-TC-spanner for the resulting graph and make it 1-good, with a logarithmic blowup in the number of edges. This means that now comparable vertices that pass through all superedges use the same type of shortcut edges; see our rerandomization lemma. Finally, given the new special form of the k-TC-spanner, we are able to directly relate its size to the size of the optimal cover of our specialized Min-Rep instance; see our rep-cover extraction lemma. This will complete the proof.

We now proceed with the formal proof.

Theorem 3.14 (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 $\mathsf{NP} \subseteq DTIME(n^{\mathrm{polylog}\, n})$.*

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. We give a reduction from MIN-REP with unrestricted parameters, whose hardness is stated in Fact 3.13.

We reduce an arbitrary MIN-REP instance on $n^{\kappa'}$ vertices to a MIN-REP instance on n vertices with parameters in the desired range (where κ' is a suitably small constant). Since MIN-REP with unrestricted parameters is $2^{\log^{1-\epsilon} n}$-inapproximable and the reduction is polynomial time, the theorem follows. The reduction consists of a sequence of five transformations on the original instance. We describe each of the transformations and specify how the parameters of the input and output MIN-REP instances are related.

1. *(disjoint copies)*
   Given an (n_0, r_0, d_0, m_0)-MIN-REP instance G_0 with OPT_0 as the solution value, $T_1(G_0, n^{\delta_1})$ is defined to be the MIN-REP instance G_1 with n^{δ_1} disjoint copies of G_0. G_1 is a (n_1, r_1, d_1, m_1)-MIN-REP instance with $n_1 = n^{\delta_1} n_0$, $r_1 = n^{\delta_1} r_0$, $d_1 = d_0$, and $m_1 = m_0$. The solution value of G_1 is $OPT_1 = n^{\delta_1} OPT_0$ because if $OPT_1 < n^{\delta_1} OPT_0$, one could, by averaging over the n^{δ_1} copies of G_0, extract a MIN-REP cover for G_0 of size smaller than OPT.
2. *(dummy vertices inside clusters)*
   Given an (n_1, r_1, d_1, m_1)-MIN-REP instance G_1 with OPT_1 as the solution value, $T_2(G_1, n^{\delta_2})$ is defined to be the MIN-REP instance G_2 obtained by increasing the size of each cluster by a factor of n^{δ_2} and not attaching any edges to the new vertices. G_2 is a (n_2, r_2, d_2, m_2)-MIN-REP instance with $n_2 = n^{\delta_2} n_1$, $r_2 = r_1$, $d_2 = d_1$, and $m_2 = n^{\delta_2} m_1$. The solution value of G_2 remains $OPT_2 = OPT_1$ because the minimum cover of G_2 does not include any isolated vertices.
3. *(blowup inside clusters with matching supergraph)*
   Given an (n_2, r_2, d_2, m_2)-MIN-REP instance G_2 with OPT_2 as the solution value, $T_3(G_2, n^{\delta_3})$ is defined to be the MIN-REP instance G_3 obtained as follows. For each cluster $\mathcal{A}_i$ in G_2, construct a supercluster $\mathcal{A}'_i$ in G_3 consisting of n^{δ_3} copies of $\mathcal{A}_i$. Let $(\mathcal{A}'_i)_k$ denote the kth copy of $\mathcal{A}_i$ inside $\mathcal{A}'_i$. Obtain $(\mathcal{B}'_i)_k$ similarly. Whenever there is an edge in G_2 between $u \in \mathcal{A}_i$ and $v \in \mathcal{B}_j$, for each $1 \le k \le n^{\delta_3}$, add an edge between the copy of u in $(\mathcal{A}'_i)_k$ and the copy of v in $(\mathcal{B}'_j)_k$. This procedure yields a (n_3, r_3, d_3, m_3)-MIN-REP instance G_3 where $n_3 = n^{\delta_3} n_2$, $r_3 = r_2$, $d_3 = d_2$, and $m_3 = m_2$. The solution value of G_3 remains $OPT_3 = OPT_2$, because given a rep-cover for G_3, if we fix some $k \in [n^{\delta_3}]$ and replace each vertex in the rep-cover with the corresponding vertex in the kth copy of the cluster inside the supercluster it belongs to, then the size of the rep-cover set becomes no larger and now it covers the underlying G_2 instance as well.
4. *(blowup inside clusters with complete supergraph)*
   Given an (n_3, r_3, d_3, m_3)-MIN-REP instance G_3 with OPT_3 as the solution value, $T_4(G_3, n^{\delta_4})$ is defined to be the MIN-REP instance G_4 obtained as follows. For each cluster $\mathcal{A}_i$ in G_3, construct a cluster $\mathcal{A}'_i$ in G_4 consisting of n^{δ_4} copies of $\mathcal{A}_i$. Let $(\mathcal{A}'_i)_k$ denote the kth copy of $\mathcal{A}_i$ inside $\mathcal{A}'_i$. Whenever there is an edge in G_3 between $u \in \mathcal{A}_i$ and $v \in \mathcal{B}_j$, for each $1 \le k_1, k_2 \le n^{\delta_4}$, add an edge between the copy of u in $(\mathcal{A}'_i)_{k_1}$ and the copy of v in $(\mathcal{B}'_j)_{k_2}$. This procedure yields a (n_4, r_4, d_4, m_4)-MIN-REP instance G_4 where $n_4 = n^{\delta_4} n_3$, $r_4 = r_3$, $d_4 = n^{\delta_4} d_3$, and $m_4 = m_3$. The solution value of G_4 remains $OPT_4 = OPT_3$, because any rep-cover for G_4 is easily seen to correspond to a rep-cover for the underlying G_3 and vice versa.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

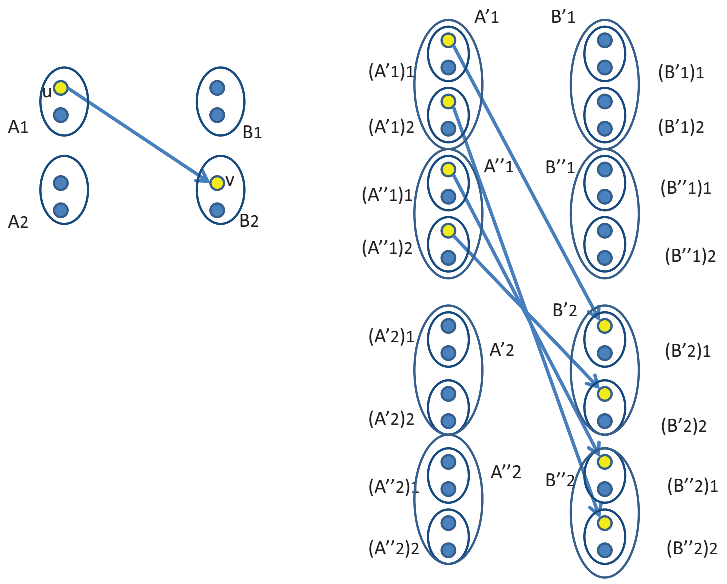

FIG. 3.5. *The basic step in the tensoring operation.*

5. *(tensoring)*
Given an (n_4, r_4, d_4, m_4)-MIN-REP instance G_4 with OPT_4 as the solution value, $T_5(G_4, n^{\delta_5})$ is defined to be the MIN-REP instance G_5 obtained by repeating the following construction $\log_2 n^{\delta_5}$ times.[3] For each cluster $\mathcal{A}_i$ in G_4, construct two clusters $\mathcal{A}'_i$ and $\mathcal{A}''_i$ in G_5. Furthermore, $\mathcal{A}'_i$ contains two copies of $\mathcal{A}_i$ and $\mathcal{A}''_i$ contains two copies of $\mathcal{A}_i$. Denote the two copies inside $\mathcal{A}'_i$ as $(\mathcal{A}'_i)_1$ and $(\mathcal{A}'_i)_2$ and similarly the two copies inside $\mathcal{A}''_i$ as $(\mathcal{A}''_i)_1$ and $(\mathcal{A}''_i)_2$. Similarly for each cluster $\mathcal{B}_j$. Now, for each edge (u, v) in G_4 with $u \in \mathcal{A}_i$ and $v \in \mathcal{B}_j$, add the following four edges in G_5: between the copy of u in $(\mathcal{A}'_i)_1$ and copy of v in $(\mathcal{B}'_j)_1$, between the copy of u in $(\mathcal{A}'_i)_2$ and copy of v in $(\mathcal{B}''_j)_2$, between the copy of u in $(\mathcal{A}''_i)_1$ and copy of v in $(\mathcal{B}''_j)_1$, and between the copy of u in $(\mathcal{A}''_i)_2$ and copy of v in $(\mathcal{B}'_j)_2$. See Figure 3.5.
The procedure yields a (n_5, r_5, d_5, m_5)-MIN-REP instance G_5 where $n_5 = n^{2\delta_5} n_4$, $r_5 = n^{\delta_5} r_4$, $d_5 = d_4$, and $m_5 = m_4$. Also, we argue that $OPT_5 = n^{2\delta_5} OPT_4$. Clearly, $OPT_5 \le n^{2\delta_5} OPT_4$ because one could choose copies of the vertices in the cover for G_4 in each of the n^{δ_5} copies of the clusters of G_4. For the other direction, notice that G_5 contains $n^{2\delta_5}$ vertex disjoint copies of G_4, and so, if $OPT_5 < n^{2\delta_5} OPT_4$, then by averaging, there would be a copy of G_4 covered using less than OPT vertices, a contradiction.

For some positive κ' sufficiently smaller than κ, we consider an arbitrary $(n^{\kappa'}, r_0, d_0, m_0)$-MIN-REP instance G_0 with optimum OPT_0, where the only constraints on the parameters are nontriviality conditions: $r_0 \in [1, n^{\kappa'}]$, $d_0 \in [1, n^{\kappa'}]$, $m_0 \in [1, n^{\kappa'}]$, and $OPT_0 \in [1, 2n^{\kappa'}]$. Let $G = T_5(T_4(T_3(T_2(T_1(G_0, n^{\delta_1}), n^{\delta_2}), n^{\delta_3}), n^{\delta_4}), n^{\delta_5})$. We choose $\delta_1, \delta_2, \delta_3, \delta_4, \delta_5$ such that G is a (n, r, d, m)-MIN-REP instance

[3]For simplicity, we assume n^{δ_5} is a power of 2.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

with $r \in [n^R, n^{R+\kappa'}]$, $d \in [n^D, n^{D+\kappa'}]$, $m \in [n^M, n^{M+\kappa'}]$, and $OPT \in [n^F, n^{F+\kappa'}]$. By definitions of the transformations, $n = n^{\kappa'+\delta_1+\delta_2+\delta_3+\delta_4+2\delta_5}$, $r \in [n^{\delta_1+\delta_5}, n^{\kappa'+\delta_1+\delta_5}]$, $d \in [n^{\delta_4}, n^{\delta_4+\kappa'}]$, $m \in [n^{\delta_2}, n^{\kappa'+\delta_2}]$, and $OPT \in [n^{\delta_1+2\delta_5}, n^{\kappa'+\delta_1+2\delta_5}]$. Therefore, choose $\delta_4 = D$, $\delta_2 = M$, $\delta_5 = F - R$, and $\delta_1 = 2R - F$. All of these values are in $(0,1)$ by restriction of the parameters in the theorem statement. Now, since $\kappa'+\delta_1+\delta_2+\delta_3+\delta_4+2\delta_5 = D+M+F+\delta_3+\kappa'$ and since $D+M+F < 1$ and κ' can be made as small as we want, we can choose $\delta_3 \in (0,1)$ such that $n = n^{\kappa'+\delta_1+\delta_2+\delta_3+\delta_4+2\delta_5}$. Therefore, G is a Min-Rep instance with parameters in the desired range. □

The variant of Min-Rep in Theorem 3.14 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 will become more clear as the proof proceeds). 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. We will construct the specialized Min-Rep instance $\mathcal{I}$ in two steps. First, we instantiate Theorem 3.14 with a set of very carefully selected parameters in order to obtain an instance $\mathcal{I}_0$, and then we apply transformation T_4 from the proof of Theorem 3.14 to $\mathcal{I}_0$. This will result in a specialized Min-Rep instance $\mathcal{I}$ that is as hard to approximate as the initial Min-Rep instance.

Toward the first step, set $\delta = \frac{k-1}{k-\frac{1}{4}}$, $\eta = \frac{\delta}{2(4k-4)(4k-2)}$, and $\zeta = \delta(\frac{4k-5}{4k-4} + \frac{1}{4k-2})$. Let κ be a sufficiently small positive constant which will be chosen in the course of the proof. 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 3.14 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 $\mathsf{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{polylog}\, n})$. The conditions on the parameters in Theorem 3.14 are satisfied since $\zeta \in (\frac{\delta}{2}, \delta)$ and $\eta + 2\eta + \zeta < \delta$.

We transform $\mathcal{I}_0$ to a specialized (n, r, d, m)-Min-Rep instance $\mathcal{I}$ by applying on $\mathcal{I}_0$ the transformation T_4. More precisely, set $\mathcal{I} = T_4(\mathcal{I}_0, n^{1-\delta})$. By definition of T_4, graph $\mathcal{I}$ has n vertices, $r = r_0$, $d = d_0 n^{1-\delta}$, and $m = m_0$. The transformation results in a bipartite graph $\mathcal{I}$ with nodes partitioned into clusters $\mathcal{A}_1, \dots, \mathcal{A}_r$ on the left, and $\mathcal{B}_1, \dots, \mathcal{B}_r$ on the right. We summarize the structural details of $\mathcal{I}$ next:

1. 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 $s \in [n^{1-\delta}]$.
2. Each group $A_{i,s}$ and $B_{j,s}$, for $i, j \in [r]$, $s \in [n^{1-\delta}]$, is a copy of $\mathcal{A}_i^0$ and, respectively, $\mathcal{B}_j^0$, from the original instance $\mathcal{I}_0$. Thus $|A_{i,s}| = |B_{j,s}| = \frac{n^\delta}{r}$.
3. For each edge (u, v) with $u \in \mathcal{A}_i^0$ and $v \in \mathcal{B}_j^0$ 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 $k_1, k_2 \in [n^{1-\delta}]$.
4. The solution value of $\mathcal{I}$ remains OPT_0 because the supergraph corresponding to $\mathcal{I}_0$ and $\mathcal{I}$ are identical.

This completes the description of the specialized Min-Rep instance $\mathcal{I}$.

An immediate consequence of our construction above is the following result.

Corollary 3.15. *The specialized* (n, r, d, m)-Min-Rep *problem defined above is* $2^{\log^{1-\epsilon} n}$*-inapproximable unless* $\mathsf{NP} \subseteq \mathit{DTIME}(n^{\mathrm{polylog}\, n})$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

From specialized MIN-REP *to the underlying* k-TC-SPANNER *instance.* As previously mentioned, to obtain our final k-TC-SPANNER instance we will analyze the graph obtained by attaching generalized butterflies to the left of the specialized MIN-REP instance $\mathcal{I}$, and broom graphs to the right of $\mathcal{I}$.

More exactly, from $\mathcal{I}$, we construct a graph $\mathcal{G}$ of diameter $k+2$ as follows. First recall that, by Definition 3.1, a generalized butterfly of diameter $k-1$ and width n' is a digraph whose vertices are identified with coordinates $[(n')^{\frac{1}{k-1}}]^{k-1} \times [k]$, and whose edges are between vertices $u = (u_1, \dots, u_{k-1}, i)$ and $v = (v_1, \dots, v_{k-1}, i+1)$ iff for all $j \neq i$, $u_j = v_j$. A vertex $(u_1, \dots, u_k, i)$ is in strip i. We first attach a disjoint generalized butterfly of diameter $k-1$ and width $n' = \frac{n^\delta}{r} = |A_{i,s}|$, 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 $BF^j(\mathcal{A}_i) = \cup_s BF^j(A_{i,s})$. Recall that the vertices in the butterfly $BF(A_{i,s})$ at distance x from $A_{i,s}$ form the xth *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 $BR(B_{i,s})$ (as defined right after the statement of Fact 3.13.) 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}.

Details on attaching butterflies. We further discuss the way the butterflies are attached to the groups. Recall that we identify vertices in $A_{i,s}$ with the last strip $BF^k(A_{i,s})$ of a disjoint butterfly for all $i \in [r]$, $s \in [n^{1-\delta}]$. Also, recall that $\mathcal{I}_0$ is a (n_0, r, d_0, m)-MIN-REP instance with $n_0 = n^\delta, r \in [n^{\delta/2}, n^{\delta/2+\kappa}], d_0 \in [n^\eta, n^{\eta+\kappa}]$, and $m \in [n^{2\eta}, n^{2\eta+\kappa}]$. Thus, for each group $A_{i,s}$ there are at most $\frac{n^\delta}{rm}$ nonisolated vertices. We will attach the butterfly $BF(A_{i,s})$ in such a way that each vertex in $BF^{k-1}(A_{i,s})$ is adjacent to at most $\frac{d_*}{m}$ nonisolated vertices in $A_{i,s}$, out of a total out-degree of size d_*. This is the crucial property exploited later in the proof. We can achieve this property in the following way. Recall that each vertex of $BF^j(A_{i,s})$ is labeled $(a_1, \dots, a_{k-1}, j)$, where $a_l \in [d_*]$ for all $l \in [k-1]$, $j \in [k]$, and each vertex $v' = (a_1, \dots, a_{k-2}, a'_{k-1}, k-1)$ connects to $v = (a_1, \dots, a_{k-2}, a_{k-1}, k)$. Thus, for a fixed prefix $b = (b_1, b_2, \dots, b_{k-2})$ all vertices $(b_1, \dots, b_{k-2}, b_{k-1}, k-1)$ connect to the same set A_b of vertices in $A_{i,s}$, and $|A_b| = d_*$ (for comparison, recall that $|A_{i,s}| = {d_*}^{k-1}$). Choose the set A_b to contain at most d_*/m nonisolated vertices, which is possible since the total fraction of nonisolated vertices in $A_{i,s}$ is $\leq \frac{1}{m}$.

A sparse TC-spanner $\mathcal{H}$ for the k-TC SPANNER *instance $\mathcal{G}$.* We first introduce a bit of notation. A k-TC-spanner for $\mathcal{G} = V_1 \cup V_2 \cup \dots \cup V_{k+3}$ is built by adding shortcut edges (u, v) between comparable u and v, where $u \in V_i, v \in V_{i+\ell}$, and $\ell \geq 2$. For given ℓ, i, we classify such a shortcut edge as *type* $\ell \& i$. Since $\mathcal{G}$ has diameter $k+2$, a k-TC-spanner of $\mathcal{G}$ remains a k-TC-spanner when a type $\ell \& i$ edge (u, v) with $\ell \geq 4$ is replaced by a type $3\&i$ edge (u, v'), where v' is a predecessor of v. Therefore, it is enough to consider k-TC-spanners with shortcut edges only of types $2\&i$ for $1 \leq i \leq k+1$ and $3\&i$ for $1 \leq i \leq k$. The k-TC-spanner constructed in Lemma 3.16 will contain only edges of type $2\&(k-2)$ and $2\&(k+1)$. Finally, we will say that the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

vertex sets X and Y are *comparable* if there exist vertices $u \in X$ and $v \in Y$ that are comparable.

LEMMA 3.16 (rep-cover spanner lemma). *There exists a k-TC-spanner $\mathcal{H}$ for $\mathcal{G}$, with $|\mathcal{H}| = O(OPT\ n^{1-\delta}(\frac{n^\delta}{r})^{\frac{2}{k-1}})$.*

Proof. We construct the graph $\mathcal{H}$ by adding some shortcut edges to $\mathcal{G}$. Let S_0 be a minimum rep-cover of $\mathcal{I}_0$ of size OPT. Recall that each $\mathcal{A}_i$ and $\mathcal{B}_j$ is replicated $n^{1-\delta}$ times in $\mathcal{I}$. Let S be the set of all replicas in $\mathcal{I}$ of vertices in S_0. Let $A_{i,j}$ and $B_{k,l}$ be two groups containing comparable pairs of vertices. Recall that $d_* = (\frac{n^\delta}{r})^{\frac{1}{k-1}}$. To get a k-TC-spanner on $BF(A_{i,j}) \cup BR(B_{k,l})$ connect each vertex v from the restriction of S to $A_{i,j}$ with all its d_*^2 comparable vertices in $BF^{k-2}(A_{i,j})$. Similarly, connect each vertex in the restriction of S to $B_{k,l}$ to its d_*^2 comparable vertices in $BR^{k+3}(B_{k,l})$. Since every vertex $u \in BF^1(A_{i,j})$ is comparable to every vertex $v \in A_{i,j}$, it follows that there is a vertex $w \in BF^{k-2}(A_{i,j})$ comparable to both u and v. Thus, between any such u and v there is a path using an edge of type $2\&(k-2)$. Similarly, every vertex in $BR^{k+3}(B_{k,l})$ is comparable to every vertex of $B_{k,l}$. By our construction, any pair of vertices $(u_1, u_{k+3}) \in BF^1(A_{i,j}) \times BR^{k+3}(B_{k,l})$ is connected by a path that uses edges of types $2\&(k-2)$ and $2\&(k+1)$. In addition, any pair of vertices $(u_1, u_{k+2}) \in BF^1(A_{i,j}) \times BR^{k+2}(B_{k,l})$, as well as $(u_2, u_{k+3}) \in BF^2(A_{i,j}) \times BR^{k+3}(B_{k,l})$ are connected by a path of length at most k using shortcut edges of types $2\&(k-2)$ and $2\&(k+1)$, respectively. By connecting all the comparable groups $A_{i,j}$ and $B_{k,l}$ in this manner, we obtain a k-TC-spanner on $\mathcal{G}$.

Since there are $n^{1-\delta}$ copies of each $A_{i,j}$ and $B_{k,l}$, the total number of shortcut edges added is $OPTn^{1-\delta}d_*^2 = OPT\ n^{1-\delta}(\frac{n^\delta}{r})^{\frac{2}{k-1}}$, and we used shortcut edges only of types $2\&(k-2)$ and $2\&(k+1)$. In addition, since $\mathcal{G}$ is transitively reduced, $\mathcal{H}$ must include all the edges of $\mathcal{G}$. We bound the size of $\mathcal{G}$ by inspecting the total number of edges in the butterflies (knd_*), the MIN-REP instance ($\leq nd$), and the brooms ($nd_* + n^{1-\delta}rd_*^2$). Thus, $|\mathcal{G}| \leq k\,r\,n^{1-\delta}\,(\frac{n^\delta}{r})^{1+\frac{1}{k-1}} + n^{2+\eta+\kappa-\delta} + n\,(\frac{n^\delta}{r})^{\frac{1}{k-1}} + n^{1-\delta}\,r\,(\frac{n^\delta}{r})^{\frac{2}{k-1}}$. The following conditions, satisfied by the parameters of our construction, suffice to show that each term of the preceding sum is, respectively, $o(|\mathcal{H}|)$. (The parameter κ is omitted from the conditions, since if the inequalities are satisfied without κ, then κ can be made sufficiently small to ensure that they are satisfied with κ.)

(3.1)
$$\zeta + (1-\delta) + \frac{2}{k-1}\left(\delta - \frac{\delta}{2}\right) > \frac{\delta}{2} + (1-\delta) + \frac{k}{k-1}\left(\delta - \frac{\delta}{2}\right) \qquad \text{or } \zeta > \delta\frac{2k-3}{2(k-1)}$$

(3.2)
$$\zeta + (1-\delta) + \frac{2}{k-1}\left(\delta - \frac{\delta}{2}\right) > 2 + \eta - \delta \qquad \text{or } \zeta > 1 + \eta - \frac{\delta}{k-1}$$

(3.3)
$$\zeta + (1-\delta) + \frac{2}{k-1}\left(\delta - \frac{\delta}{2}\right) > 1 + \frac{1}{k-1}\left(\delta - \frac{\delta}{2}\right) \qquad \text{or } \zeta > \delta\frac{2k-3}{2(k-1)}$$

(3.4)
$$\zeta + (1-\delta) + \frac{2}{k-1}\left(\delta - \frac{\delta}{2}\right) > (1-\delta) + \frac{\delta}{2} + \left(\delta - \frac{\delta}{2}\right)\frac{2}{k-1} \qquad \text{or } \zeta > \frac{\delta}{2} \qquad \square$$

Path analysis and rerandomization. The next lemma shows that the k-TC-spanner $\mathcal{H}$ defined above and analyzed in Lemma 3.16 is nearly optimal.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA 3.17. *Any k-TC-spanner $\mathcal{K}$ of $\mathcal{G}$ has*

$$|\mathcal{K}| = \Omega\left(OPT n^{1-\delta}\left(\frac{n^{\delta}}{r}\right)^{\frac{2}{k-1}} / \log n\right).$$

Proof. Given a k-TC-spanner $\mathcal{K}$ of $\mathcal{G}$ with $o(\frac{n^{1-\delta}d_*^2}{\log n})OPT$ edges, we show that we can construct a MIN-REP cover for $\mathcal{I}$ of size $o(OPT)$, which is a contradiction (recall that $d_* = (\frac{n^\delta}{r})^{\frac{1}{k-1}}$). We will accomplish this by a series of transformations which modify $\mathcal{K}$ into a k-TC-spanner that uses shortcut edges only of the type $2\&(k-2)$ and $2\&(k+1)$. The process increases the size of the k-TC-spanner only by a logarithmic factor. Finally, we show that from the modified k-TC-spanner, one can extract a MIN-REP cover of size $o(OPT)$ for $\mathcal{I}$, the desired contradiction.

We call a superedge $(\mathcal{A}_i, \mathcal{B}_j)$, where $i, j \in [r]$, *deletable with respect to* $\mathcal{K}$ if at least $1/4$ of the vertex pairs $(u, v) \in BF^1(\mathcal{A}_i) \times BR^{k+3}(\mathcal{B}_j)$ have a path between them in $\mathcal{K}$ of length at most k, and which does not contain edges both of type $2\&(k-2)$ and $2\&(k+1)$. Our first step is to show that such cluster pairs can be essentially ignored.

LEMMA 3.18 (path analysis lemma). *The number of deletable superedges with respect to $\mathcal{K}$ is $o(OPT)$.*

Proof. We call a path *canonical* if it contains shortcut edges of types both $2\&(k-2)$ and $2\&(k+1)$; otherwise, a path is *alternative*. Observe that any alternative path contains at least one shortcut edge from among the following three cases: (1) shortcut edges crossing both V_k and V_{k+1}, i.e., one of the shortcut edge types: $3\&(k-2)$, $3\&(k-1)$, $2\&(k-1)$, $2\&k$, and $3\&k$; (2) shortcut edges of type $3\&\ell$, where $\ell \le k-3$; (3) shortcut edges of type $2\&\ell$, where $\ell \le k-3$. Let S_B be the set of all the shortcut edge types contained in the above three cases. Then $|S_B| = \Theta(k)$. Now, for each shortcut edge type $S \in S_B$, let $Del(S) = \{(i, j) \in [r]^2|$ at least $\frac{1}{4|S_B|}$ fraction of pairs $(u, v) \in BF^1(\mathcal{A}_i) \times BR^{k+3}(\mathcal{B}_j)$ have an alternative path containing a shortcut edge of type $S\}$. By a union bound, the total number of deletable superedges is at most $\sum_{S \in S_B} Del(S)$. Hence, it suffices to show that for all $S \in S_B$, $Del(S) = o(OPT)$.

Let $C(S) = \{(u, v) \in \cup_{(i,j)\in[r]^2}(BF^1(\mathcal{A}_i) \times BR^{k+3}(\mathcal{B}_j)) \mid \exists$ an alternative path between u and v containing a shortcut edge of type $S\}$. By the definition of $Del(S)$, since for all $i \in [r]$, $|BF^1(\mathcal{A}_i)| = \frac{n}{r}$ and $|BR^{k+3}(B_i)| = n^{1-\delta}d_*^2$, we have

$$|C(S)| \ge |Del(S)| \frac{1}{4|S_B|} \frac{n}{r}\ n^{1-\delta}\ d_*^2. \tag{3.5}$$

Now we will obtain upper bounds of $|C(S)|$ in terms of OPT for each of three cases of shortcut edges, thus obtaining upper bounds on $Del(S)$. Recall that $\delta = \frac{k-1}{k-\frac{1}{4}}$, $\eta = \frac{\delta}{2(4k-4)(4k-2)}$, and $\zeta = \delta(\frac{4k-5}{4k-4} + \frac{1}{4k-2})$. Also, recall $r \in [n^{\frac{\delta}{2}}, n^{\frac{\delta}{2}+\kappa}]$, $d \in [n^{(1-\delta)+\eta}, n^{(1-\delta)+\eta+\kappa}]$, and $m \in [n^{2\eta}, n^{2\eta+\kappa}]$ for some small enough constant κ and $d_* = (\frac{n^\delta}{r})^{1/(k-1)}$. We mostly ignore κ below since we can make it as small a constant as we like.

Suppose that S is a shortcut edge type from the first case. Then S is a shortcut of type $\ell_1\&(k-\ell_2)$, where $2 \le \ell_1 \le 3, 0 \le \ell_2$, and $\ell_1 - \ell_2 \ge 1$. Now we obtain that for any shortcut of type S, the shortcut can be used for at most $d_*^{k-1-\ell_2} d_*^{3+\ell_2-\ell_1} = d_*^{k+2-\ell_1}$ pairs $(u, v) \in C(S)$. Hence, $|C(S)| \le d_*^{k+2-\ell_1} \cdot OPT\frac{n^{1-\delta}d_*^2}{\log n} \le d_*^k \cdot OPT\frac{n^{1-\delta}\ d_*^2}{\log n}$. From (3.5), we obtain that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$|Del(S)| \leq 4|S_B| \frac{d_*^k OPT \; n^{1-\delta} \; d_*^2}{\frac{n}{r} \; n^{1-\delta} \; d_*^2 \log n} = O\left(\frac{d_*}{n^{1-\delta}\log n}\right) OPT.$$

Then because $\delta < \frac{k-1}{k-\frac{1}{2}}$, it follows that $1-\delta > \frac{\delta}{2(k-1)}$, and so we obtain that $n^{1-\delta}$ is a polynomial factor larger than $d_* = (\frac{n^\delta}{r})^{1/(k-1)}$, which proves that $|Del(S)| = o(OPT)$.

Now suppose that S is a shortcut type of the second case. Let S be type $3\&\ell$, where $1 \leq \ell \leq k-3$. Now, from the fact that out-degree of each vertex in V_k is at most $n^{1-\delta+\eta+\kappa}$, we obtain that for any shortcut of type S, the shortcut can be used for at most $d_*^{\ell-1} d_*^{k-3-\ell} n^{(1-\delta)+\eta} d_*^2 = d_*^{k-4} n^{(1-\delta)+\eta} d_*^2$ many pairs $(u,v) \in C(S)$ (ignoring κ as mentioned above). Hence, up to small polynomial factors,

$$(3.6) \qquad |C(S)| \leq d_*^{k-4} n^{(1-\delta)+\eta} d_*^2 \cdot OPT \frac{n^{1-\delta} d_*^2}{\log n} = \frac{d_*^k n^{2-2\delta+\eta}}{\log n} OPT.$$

From (3.5) and (3.6), we obtain $|Del(S)| \leq 4|S_B| \frac{n^\eta}{d_* \log n} OPT$. Now, $\eta < \frac{\delta}{2(k-1)}$, and so, $|Del(S)| = o(OPT)$.

Now suppose that S is a shortcut type of the third case. Let S be type $2\&\ell$, where $\ell \leq k-3$. Note that for any vertex v in V_{k-1}, the number of nonisolated vertices in V_k that v is connected to is $\frac{d_*}{m}$. Hence, together with the fact that the out-degree of each vertex in V_k is at most $n^{1-\delta+\eta+\kappa}$, we obtain that for any shortcut of type S, the shortcut can be used for at most $\frac{d_*^{k-3}}{n^{2\eta}} n^{(1-\delta)+\eta} d_*^2$ many pairs (u,v) in $C(S)$ (up to small polynomial factors). Then

$$(3.7) \qquad |C(S)| \leq \frac{d_*^{k-3}}{n^{2\eta}} n^{(1-\delta)+\eta} d_*^2 \cdot OPT \frac{n^{1-\delta} d_*^2}{\log n} = \frac{d_*^{k+1} n^{2-2\delta+\eta}}{n^{2\eta}\log n} OPT.$$

From (3.5), (3.7), and the fact that $n^\eta = o(n^{2\eta}\log n)$, we get $|Del(S)| \leq 4|S_B| \frac{n^\eta}{n^{2\eta}\log n}$ $OPT = o(OPT)$. ☐

Let S_B be the set of all shortcut edge types included in the three cases. We analyze the three cases separately and show that for each $S \in S_B$, the number of superedges (A_i, B_j), $(i,j) \in [r]^2$, such that at least a $\frac{1}{4|S_B|}$ fraction of pairs $(u,v) \in BF^1(\mathcal{A}_i) \times BR^{k+3}(\mathcal{B}_j)$ have an alternative path containing a shortcut of type S, is $o(OPT)$. Then by a union bound over $S \in S_B$, we prove the lemma. The analysis of case (1) relies on the fact that the degree of each nonisolated vertex of V_k is at least $n^{1-\delta} \geq d_*$. For case (2), we need the facts that the out-degree of each vertex in V_k is at most $d_0 n^{1-\delta}$ and that $n^\eta = o(d_*)$. For case (3), we use the facts that every vertex v in V_{k-1} is connected to is at most $\frac{d_*}{m}$ nonisolated vertices in V_k, and that $n^\eta = o(n^{2\eta})$.

Next, form the graph $\mathcal{G}'$ from $\mathcal{G}$ by deleting all edges of $\mathcal{G}$ connecting $\mathcal{A}_i$ to $\mathcal{B}_j$, for all the deletable superedges $(\mathcal{A}_i, \mathcal{B}_j)$ with respect to $\mathcal{K}$. Similarly, obtain a graph $\mathcal{K}'$ from $\mathcal{K}$ as follows: for all deletable superedges $(\mathcal{A}_i, \mathcal{B}_j)$ with respect to $\mathcal{K}$, delete all edges of $\mathcal{K}$ connecting $\mathcal{A}_i$ to $\mathcal{B}_j$, and also delete all shortcuts in $\mathcal{K}$ of types other than $2\&(k-2)$ and $2\&(k+1)$. Note that for any cluster pair $(\mathcal{A}_i, \mathcal{B}_j)$ of $\mathcal{G}'$, either there are no edges between vertices in $\mathcal{A}_i$ and $\mathcal{B}_j$ or at least $\frac{3}{4}$ of the pairs in $BF^1(\mathcal{A}_i) \times BR^{k+3}(\mathcal{B}_j)$ are connected by a canonical path. Also define a Min-Rep instance $\mathcal{I}'$ from $\mathcal{I}$ by deleting all edges in $\mathcal{I}$ corresponding to all the deletable superedges with respect to $\mathcal{K}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

For $\mu \in [0,1]$, we say a subgraph of $TC(\mathcal{G})$ is a *μ-good k-TC-spanner for $\mathcal{G}$* if for every $(i,j) \in [r]^2$ such that $\mathcal{A}_i$ and $\mathcal{B}_j$ are comparable in $\mathcal{G}$, at least a μ fraction of pairs $(u,v) \in BF^1(\mathcal{A}_i) \times BR^{k+3}(\mathcal{B}_j)$ are connected by canonical paths in the subgraph. For example, the graph $\mathcal{K}'$ is a $\frac{3}{4}$-good k-TC-spanner for $\mathcal{G}$.

LEMMA 3.19 (rerandomization lemma). *If a $\frac{3}{4}$-good k-TC-spanner $\mathcal{K}'$ for $\mathcal{G}'$ is given, then there exists $\mathcal{K}''$, a 1-good k-TC-spanner for $\mathcal{G}'$, such that $|\mathcal{K}''| \le O(|\mathcal{K}'| \cdot \log n)$.*

Proof. First, we fix some notation. Consider some $(i,j) \in [r]^2$ such that there is an edge between a vertex in $\mathcal{A}_i$ and a vertex in $\mathcal{B}_j$ in $\mathcal{G}'$. Let $S_{i,j}$ be the set of vertices in $\mathcal{A}_i$ that are adjacent to $\mathcal{B}_j$, and let $T_{i,j}$ be the set of vertices in $\mathcal{B}_j$ that are adjacent to $\mathcal{A}_i$. We know that at least $\frac{3}{4}$ of the vertices in $BF^1(\mathcal{A}_i)$ have a path that uses an edge of type $2\&(k-2)$ to $S_{i,j}$ and at least $\frac{3}{4}$ of the vertices in $BR^{k+3}(\mathcal{B}_j)$ have a path of length 1 from $T_{i,j}$. By a Markov argument, for at least $\frac{1}{2}$ of the groups $A_{i,s}$ in $\mathcal{A}_i$, at least $\frac{1}{2}$ of the vertices in $BF^1(A_{i,s})$ must have a path that uses an edge of type $2\&(k-2)$ to $S_{i,j}$. Call the butterfly attached to such a group $A_{i,s}$ an *(i,j)-good butterfly*, and call the set of vertices in $BF^{k-2}(A_{i,s})$ that have shortcut edges to $S_{i,j}$ *(i,j)-helpful vertices*. Similarly, for at least $\frac{1}{2}$ of the groups $B_{j,t}$, at least $\frac{1}{2}$ of the vertices in $BR^{k+2}(B_{j,t})$ have shortcut edges from $T_{i,j}$. We call the brooms attached to such groups $B_{j,t}$ *(i,j)-good brooms*, and we call the vertices in $BR^{k+2}(B_{j,t})$ that have shortcut edges to $T_{i,j}$ *(i,j)-helpful vertices*. It will be clear from context whether a helpful vertex is to the left or to the right of the MIN-REP instance.

Our construction of $\mathcal{K}''$ ensures that in $\mathcal{K}''$, for any two comparable clusters $(\mathcal{A}_i, \mathcal{B}_j)$, each vertex in $BF^1(\mathcal{A}_i)$ is comparable to a helpful vertex in $BF^{k-2}(\mathcal{A}_i)$, and each vertex in $BR^{k+3}(\mathcal{B}_j)$ is helpful. This is enough to ensure that $\mathcal{K}''$ is a 1-good k-TC-spanner for $\mathcal{G}'$. We will construct $\mathcal{K}''$ to be equal to $\bigcup_{r=1}^{O(\log n)} \Pi_r(\mathcal{K}')$, where each Π_r is a random transformation of $\mathcal{K}'$ that moves the shortcut edges.

Each Π_r will be the composition of several transformations on the edges of $\mathcal{K}'$. The transformations move only shortcut edges, but not transitive reduction edges, in $\mathcal{K}'$. Informally, the first transformation randomly permutes the groups in each cluster on the left side of the MIN-REP instance, the second randomly permutes the groups in each cluster of the right side of the MIN-REP instance, the third randomly permutes the edges of the butterfly graph, and the fourth randomly permutes the broomsticks. Formally, we have the following:

- *Left group permutations: Π^{lg}.*
  For each $i \in [r]$, independently choose a random permutation $\pi_i : [n^{1-\delta}] \to [n^{1-\delta}]$. For each cluster $\mathcal{A}_i$, if (u,v) is an edge in $\mathcal{K}'$ with $u, v \in BF(A_{i,s})$, then there is an edge (u',v') in $\Pi^{lg}(\mathcal{K}')$, where u' and v' are the copies of u and v, respectively, in $BF(A_{i,\pi_i(s)})$.
- *Right group permutations: Π^{rg}.*
  For each $j \in [r]$, independently choose a random permutation $\pi_j : [n^{1-\delta}] \to [n^{1-\delta}]$. For each cluster $\mathcal{B}_j$, if (u,v) is an edge in $\mathcal{K}'$ with $u, v \in BR(B_{j,s'})$, then there is an edge (u',v') in $\Pi^{rg}(\mathcal{K}')$, where u' and v' are the copies of u and v, respectively, in $BR(B_{j,\pi_j(s')})$.
- *Butterfly permutations: Π^{bf}.*
  For each $i \in [r]$ and $s \in [n^{1-\delta}]$, label a vertex u in $BF(A_{i,s})$ as $(a_1, a_2, \dots, a_{k-1}, m) \in [d_*]^{k-1} \times [k]$, where $u \in V_m$ and $(a_1, a_2, \dots, a_{k-1})$ is the usual vertex labelling that defines a generalized butterfly graph. Now, for every (i,s) and every $(a_1, \dots, a_{k-3}) \in [d_*]^{k-3}$, independently choose two random permutations $\pi_{i,s}^{(a_1,\dots,a_{k-3})} : [d_*] \to [d_*]$ and $\sigma_{i,s}^{(a_1,\dots,a_{k-3})} : [d_*] \to [d_*]$. For any edge

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$(u, w) \in BF^{k-2}(A_{i,s}) \times BF^k(A_{i,s})$, where $u = (a_1, \dots, a_{k-3}, a_{k-2}, a_{k-1}, k-2)$ and $w = (a_1, \dots, a_{k-3}, a'_{k-2}, a'_{k-1}, k)$, there exists the edge (u', w) in $\Pi^{bf}(\mathcal{K}')$ where $u' = (a_1, \dots, a_{k-3},\ \pi_{i,s}^{(a_1,\dots,a_{k-3})}(a_{k-2}),\ \sigma_{i,s}^{(a_1,\dots,a_{k-3})}(a_{k-1}),\ k-2)$. All other edges in the butterfly stay fixed.

- *Broom permutations:* Π^{br}.
  For each $j \in [r]$ and $s' \in [n^{1-\delta}]$, independently choose random permutations $\pi_{j,s'} : [d_*] \to [d_*]$ and $\sigma_{j,s'} : [d_*] \to [d_*]$. Label a vertex $v \in BR^{k+2}(B_{j,s'})$ as an element of $[d_*]$ and label a vertex $w \in BR^{k+3}(B_{j,s'})$ as an element of $[d_*] \times [d_*]$ in the natural way. If $(u, w) \in BR^{k+1}(B_{j,s'}) \times BR^{k+3}(B_{j,s'})$ is an edge in $\mathcal{K}'$, then $(u, w') \in BR^{k+1}(B_{j,s'}) \times BR^{k+3}(B_{j,s'})$ is an edge in $\Pi^{br}(\mathcal{K}')$, where $w' = (\pi_{j,s'}(w_1), \sigma_{j,s'}(w_2))$ if the label of w is (w_1, w_2). All other edges in the broom stay fixed.

Now, for each $r = 1, \dots, O(\log n)$, define Π_r to be the composition of Π^{lg}, Π^{rg}, Π^{bf}, and Π^{br}. For each r, choose all the permutations independently. As we said before, we set $\mathcal{K}'' = \cup_r \Pi_r(\mathcal{K}')$.

CLAIM 3.20. *For each $(i, j) \in [r]^2$ such that $\mathcal{A}_i$ and $\mathcal{B}_j$ are comparable, for any $u \in BF^1(\mathcal{A}_i)$ and $v \in BR^{k+3}(\mathcal{B}_j)$,*

$$\Pr_{\Pi_r}[u \textit{ is in a } (i,j)\textit{-good butterfly in } \Pi_r(\mathcal{K}')] \geq \frac{1}{2},$$

$$\Pr_{\Pi_r}[v \textit{ is in a } (i,j)\textit{-good broom in } \Pi_r(\mathcal{K}')] \geq \frac{1}{2}.$$

Proof. At least half the butterflies attached to $\mathcal{A}_i$ are good as discussed above, and hence, for every vertex $u \in BF^1(\mathcal{A}_i)$, the left group permutations ensure that with probability at least $\frac{1}{2}$, the edges of a good butterfly are mapped to the butterfly that u belongs to. The right group permutations provide the same function for a vertex $v \in BR^{k+3}(\mathcal{B}_j)$. ◻

CLAIM 3.21. *Fix $(i, j) \in [r]^2$ such that $\mathcal{A}_i$ and $\mathcal{B}_j$ are comparable. For every vertex $v \in BR^{k+3}(\mathcal{B}_j)$,*

$$\Pr_{\Pi_r}[v \textit{ is an } (i,j)\textit{-helpful vertex } | v \textit{ in } (i,j)\textit{-good broom}] \geq \frac{1}{2}.$$

Proof. At least half of the broomsticks of a good broom are helpful (i.e., incident to a shortcut edge), and hence for every vertex $v \in BR^{k+3}(\mathcal{B}_j)$, the broom permutations ensure that with probability at least $1/2$, vertex v is incident to a shortcut edge. ◻

CLAIM 3.22. *For all $(i, j) \in [r]^2$ such that $\mathcal{A}_i$ and $\mathcal{B}_j$ are comparable, and for all $u \in BR^1(\mathcal{A}_i)$,*

$$\Pr_{\Pi_r}[u \textit{ is comparable to a } (i,j)\textit{-helpful vertex } | u \textit{ is in a } (i,j)\textit{-good butterfly}] \geq \frac{1}{2}.$$

Proof. Suppose $BF(A_{i,s})$ is a (i, j)-good butterfly. For any $(a_{k-2}, a_{k-1}) \in [d_*]^2$, let $S_{(a_{k-2},a_{k-1})}$ be the set of vertices u in $BF^1(A_{i,s})$ such that u is labeled as $(a_1, \dots, a_{k-3}, a_{k-2}, a_{k-1}, 1)$, where $(a_1, \dots, a_{k-3})$ are arbitrary elements of $[d_*]^{k-3}$. Note that all the vertices in a given $S_{(a_{k-2},a_{k-1})}$ are comparable to the same set of vertices in $BF^{k-2}(A_{i,s})$ and hence, either they are all comparable to an (i, j)-helpful vertex or none of them are. Hence, at least $\frac{1}{2}$ of the $S_{(a_{k-2},a_{k-1})}$'s must have every vertex comparable to a helpful vertex in $BF^{k-2}(A_{i,s})$. Now, because of the random

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

butterfly permutations, a given vertex $v \in BF^1(A_{i,s})$ falls in such a $S_{(a_{k-2},a_{k-1})}$ with probability at least $\frac{1}{2}$. ∎

Thus, for two vertices u and v in comparable $\mathcal{A}_i$ and $\mathcal{B}_j$, respectively, the probability that u and v are connected by a canonical path in $\Pi_r(\mathcal{K}')$ is at least $\frac{1}{16}$. Since we take $O(\log n)$ independent random transformations Π_r, the probability that u and v will be connected by a canonical path in at least one $\Pi_r(\mathcal{K}')$ is at least $1 - \frac{1}{\mathrm{poly}(n)}$. Taking a union bound over all vertex pairs in $\mathcal{A}_i$ and $\mathcal{B}_j$ as well as all possible i and j, we find that with probability at least $\frac{1}{2}$, $\mathcal{K}''$ has a canonical path between any comparable $u \in V_1$ and $v \in V_{k+3}$. Therefore, the desired $\mathcal{K}''$ exists and is of size $O(|\mathcal{K}'| \cdot \log n)$. ∎

Now that the k-TC-spanner is 1-good, it is easier to reason about rep-covers of the underlying MIN-REP instance. We show below in Lemma 3.23 that there exists a rep-cover for $\mathcal{I}$ of size $o(OPT)$. This is a contradiction, completing the proof of Lemma 3.17. ∎

LEMMA 3.23 (rep-cover extraction lemma). *Given $\mathcal{K}''$, a 1-good k-TC-spanner for $\mathcal{G}'$, of size $o(OPT \cdot n^{1-\delta} \cdot d_*^2)$, there exists a* MIN-REP *cover of $\mathcal{I}$ of size $o(OPT)$.*

Proof. For $s \in [n^{1-\delta}]$, define $\mathcal{K}''_s$ to be the subgraph of $\mathcal{K}''$ induced by $\cup_{i=1}^r \big(BF(A_{i,s}) \cup BR(B_{i,s})\big)$. The $\mathcal{K}''_s$ are clearly disjoint. By averaging, there exists an $\bar{s}$ such that $|\mathcal{K}''_{\bar{s}}| \le o(OPT \cdot d_*^2)$.

We further partition the shortcut edges in $\mathcal{K}''_{\bar{s}}$ into d_*^2 parts. For each $x, y \in [d_*]$, let $U_{x,y}$ denote the set of all the nodes in $\cup_{i=1}^r BF^1(A_{i,\bar{s}})$ with butterfly coordinates $(u_1, \dots, u_{k-2}, x, y, 1)$, where $u_1, \dots, u_{k-2} \in [d_*]$. To partition the corresponding broomsticks, identify the nodes in $BR^{k+2}(B_{i,s})$ with $[d_*]$, and for each such node $x \in [d_*]$, identify its descendants in $BR^{k+3}(B_{i,s})$ with $(x,1), \dots, (x, d_*)$. For each $x, y \in [d_*]$, let $U'_{x,y}$ denote the set of all the broomsticks $\cup_{i=1}^r BR^{k+3}(B_{i,\bar{s}})$ with coordinates (x,y). Define $\mathcal{K}''_{\bar{s},x,y}$ to be the subgraph of $\mathcal{K}''_{\bar{s}}$ induced by the nodes comparable to the nodes in $U_{x,y} \cup U'_{x,y}$.

Observe that the shortcut edges in different $\mathcal{K}''_{\bar{s},x,y}$ are disjoint because (a) different $U'_{x,y}$ are disjoint and (b) the descendants in V_{k-2} of different $U_{x,y}$ are also disjoint. Thus, by averaging, there exist $\bar{x}, \bar{y}$ such that $\mathcal{K}''_{\bar{s},\bar{x},\bar{y}}$ contains $o(OPT)$ shortcut edges.

Let S be the set of vertices in V_k and V_{k+1} that are incident to shortcut edges in $\mathcal{K}''_{\bar{s},\bar{x},\bar{y}}$. Then $|S| \le o(OPT)$. Observe that S is a rep-cover for the MIN-REP instance $\mathcal{I}'_{\bar{s}}$ obtained by restricting $\mathcal{I}'$ to the edges between $A_{i,\bar{s}}$ and $B_{j,\bar{s}}$. This holds because in $\mathcal{K}''_{\bar{s}}$, each comparable pair of nodes in $U_{\bar{x},\bar{y}} \times U'_{\bar{x},\bar{y}}$ is connected by a canonical path. But a MIN-REP cover for $\mathcal{I}'_{\bar{s}}$ is also a a MIN-REP cover for $\mathcal{I}'$ by definition of $\mathcal{I}$. Finally, given a rep-cover S of $\mathcal{I}'$, we can get a rep-cover of $\mathcal{I}$ by adding at most two vertices per superedge deleted from $\mathcal{I}$ to obtain $\mathcal{I}'$. Since $o(OPT)$ superedges were deleted and since $|S| \le o(OPT)$, we obtain a MIN-REP cover for $\mathcal{I}$ of size $o(OPT)$. ∎

3.3. NP-hardness of k-TC-SPANNER.

Theorem 3.2 breaks down for large k, specifically, $k = \Omega(\lg n / \lg \lg n)$. In this setting, we have the following.

THEOREM 3.24. *For all $\epsilon > 0$ and all $k < n^{1-\epsilon}$, it is NP-hard to approximate the size of the sparsest k-TC-spanner within a factor of $1 + \gamma$, for some $\gamma = \Omega\left(\frac{1}{k}\right)$.*

Proof. We reduce from 3NODECOVER. An instance of 3NODECOVER consists of a collection D of subsets of a universe X. Each subset contains at most three elements, and each element of X is contained in at most two subsets. The goal is to output a minimum size subcollection of D, such that the union of the sets in the subcollection is X. We use the following result by Berman and Karpinski [15].

THEOREM 3.25 (from [15]). 3NODECOVER *is NP-hard to approximate within a factor of $1 + c$, for some constant $c > 0$.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

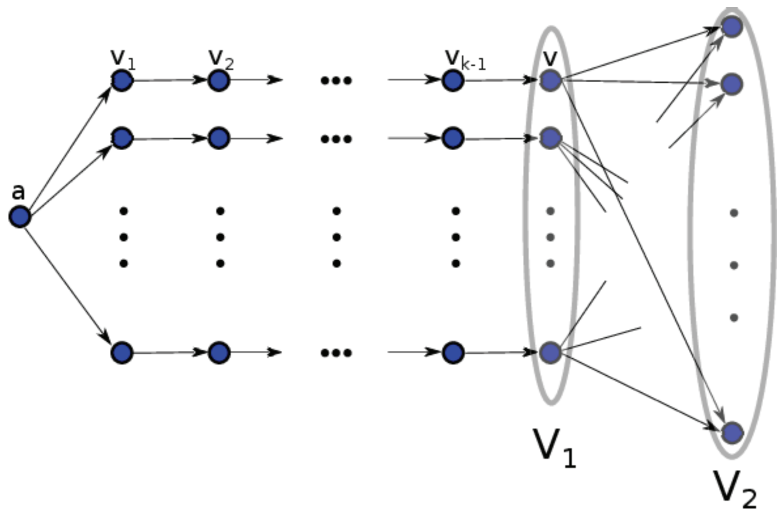

FIG. 3.6. *Reduction from* 3NODECOVER *to* k-TC-SPANNER.

We now give a reduction from 3NODECOVER to k-TC-SPANNER. For a given instance R of 3NODECOVER, we construct the following graph G, depicted in Figure 3.6. Let V_1 be the set of vertices representing each set $d \in D$. Let V_2 be the set of vertices representing each element $t \in X$. Draw a directed edge from each vertex in V_1 corresponding to $d \in D$ to the vertices in V_2 corresponding to elements of d. Add an extra vertex a. For each vertex $v \in V_1$, add $k-1$ new vertices $v_1, v_2, \ldots, v_{k-1}$ and connect them via a directed path of length k passing through $a, v_1, v_2, \ldots, v_{k-1}, v$ in the given order. Call this path $P(v)$.

Let OPT_S be the size of a sparsest k-TC-spanner of G, and let OPT_{3NC} be the size of the solution to the initial instance $R = (D, X)$ of 3NODECOVER. Let $|D| = n$.

CLAIM 3.26. $OPT_S = OPT_{3NC} + kn + \sum_{d \in D} |d|$.

Proof. All edges of G need to be included in a TC-spanner because G is transitively reduced. The number of edges in G is $kn + \sum_{d \in D} |d|$, where the first term accounts for the edges on the n paths $P(v)$, and the second, for the edges from V_1 to V_2.

Recall that *shortcuts* are edges from the transitive closure of G, added to G to obtain a TC-spanner. We show that there is a sparsest spanner H of G that contains only shortcuts from vertex a to some vertices in V_1.

For $v \in V_1$, suppose that H contains an edge (v_i, t) for some $v_i \in P(v)$ and $t \in V_2$. We claim that such an edge (v_i, t) can be replaced with an edge (a, u), where $u \in V_1$ and u is adjacent to t. Indeed, all vertices on $P(v)$, besides a, are already at distance at most k from t in G. Thus, to reach t from a it is enough to include in H a shortcut from a to any $u \in V_1$ that is adjacent to t. Similarly, an edge e between vertices on a path $P(v)$ in H can be replaced by an edge (a, v). Indeed, such an edge e can only be useful to connect a to some vertices in V_2, via a path that passes through v.

Among the edges from a to V_1, a sparsest k-TC-spanner need only contain the edges that connect a to a minimum set of vertices in V_1 that cover V_2. Thus, there are exactly OPT_{3NC} such edges and, by the argument above, no other shortcuts are needed. ☐

Suppose that there exist a positive γ and an algorithm $\mathcal{A}$ that approximates the size of a sparsest k-TC-spanner within $1 + \gamma$. Namely, $\mathcal{A}$ outputs s, such that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$OPT_S \leq s \leq (1+\gamma)OPT_S$. We show that $\gamma \geq \frac{c}{19+6k}$, where c is the constant from Theorem 3.25.

Each set d contained in an optimal solution to R covers at most three elements of the universe X. Therefore, $|X| \leq 3OPT_{3NC}$. Any element of X is contained in at most 2 sets of D and, therefore, $|X| \geq \frac{n}{2}$. This implies that $n \leq 6OPT_{3NC}$. Let $s' = s - (k+3)n$. Then

$$\begin{aligned} OPT_{3NC} \leq s' &\leq OPT_{3NC} + \gamma(OPT_{3NC} + kn + 3n) \\ &\leq OPT_{3NC} + \gamma(OPT_{3NC} + 6kOPT_{3NC} + 18OPT_{3NC}) \\ &= OPT_{3NC}(1 + \gamma(19+6k)). \end{aligned}$$

Finally, Theorem 3.25 implies that $\gamma \geq \frac{c}{19+6k}$. Thus, $\gamma = \Omega(\frac{1}{k})$. □

4. Sparse k-TC-spanners for path separable graphs.

The previously known TC-spanner constructions (e.g., in [42] the authors give 2-TC-spanners for planar digraphs of size $O(n^{3/2}\log n)$ using Lipton–Tarjan separators [53]) follow a divide-and-conquer approach. We cut the graph into two or more roughly equal-sized components, ensure that comparable vertices in different components have a path of length at most k between them, and then recurse on each component. In this section, we extend the divide-and-conquer approach to improve TC-spanner constructions for planar digraphs and, more generally, for graph families with a fixed forbidden minor.

For our divide-and-conquer approach, we require that the digraphs in question have "small" separators. The most useful notion of separability for our purposes is *path separability.*

DEFINITION 4.1 (a variant of [1]). *Let G be a connected undirected graph with n vertices. The graph G is (s,m)-*path separable *(for $m \geq n/2$) if for every rooted spanning tree T of G, at least one of the following holds:*

1. *there exists a set S of at most s monotone paths*[4] *in T, such that each connected component of $G\backslash S$ is of size at most m, or*
2. *for some $s' < s$, there exists a set S of s' monotone paths in T, such that the largest connected component of $G\backslash S$ is $(s-s',m)$-path separable.*

*If G is $(s,n/2)$-path separable, it is also called s-*path separable.

Suppose G is s-path separable. Then if we run the recursive process described by Definition 4.1 and fix a choice of rooted spanning tree T at each step, then we obtain a collection S of s paths in G such that each connected component of $G \setminus S$ is of size at most $n/2$. We will refer to such a collection S as an *s-path separator* of G.

Note that the total number of vertices in an s-path separator is left unspecified. Trees are 1-path separable, since S can be taken to be the centroid. Similarly, graphs of treewidth w are $(w+1)$-path separable. Thorup [71] showed that every planar graph is 3-path separable. Indeed, for every planar graph G and every rooted spanning tree for it, he proved that there exists a set of three root paths of the tree whose removal disconnects the graph into components of size at most $n/2$. Abraham and Gavoille [1] studied the more general case of H-minor-free graphs. For a fixed graph H, we say that a graph is H-minor-free if it belongs to a minor-closed graph family that does not have H as a minor.

THEOREM 4.2 (derived from Theorem 1 of [1]). *Every H-minor-free graph is s-path separable, for $s = s(H)$. Moreover, for any H-minor-free graph G and for any sequence of choices of rooted spanning trees in Definition* 4.1*, an s-path separator for G can be computed in polynomial time.*

[4] A monotone path in a rooted tree is a subpath of a path with one endpoint at the root.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The definition of path separability used by Abraham and Gavoille is slightly different from Definition 4.1. Namely, in the recursive process naturally described by our Definition 4.1, the monotone paths to be removed may be chosen from different spanning trees of G at different levels of recursion. In contrast, Abraham and Gavoille insist that all the paths come from the *same* spanning tree of G. However, if one inspects Abraham and Gavoille's proof of Theorem 4.2, it can be readily seen [43] that the separators obtained satisfy our stronger notion of path separability. The flexibility in choosing different spanning trees at different levels of recursion forms a key part of our argument.

Our main theorem in this section is the following.

THEOREM 4.3. *If G' belongs to an H-minor-free graph family (where H is a fixed minor), and if G is a directed graph whose underlying undirected graph is G', then G has a 2-TC-spanner of size $O(n \log^2 n)$ and, more generally, a k-TC-spanner of size $O(n \cdot \log n \cdot \lambda_k(n))$, where $\lambda_k(\cdot)$ is the kth-row inverse Ackermann function.*

As mentioned in section 1.2, these results are superior to previous constructions used in the access control literature [10]. Additionally, as pointed out after Lemma 1.2, Theorem 4.3 produces monotonicity testers with better query complexity than previously known for minor-closed poset families whose Hasse graphs forbid a fixed minor.

Proof of Theorem 4.3. By Theorem 4.2, graph G' is s-path separable for a constant s.

First, we describe a preprocessing step resembling that in [71] in which the digraph is divided into subgraphs so that constructing a TC-spanner of each subgraph individually results in a TC-spanner of the entire graph. Then we show how to efficiently construct sparse 2-TC-spanners of all these path separable subgraphs. Finally, we give our construction for general $k \geq 3$.

Preprocessing step. Let G be a transitively reduced, connected digraph. If G is not weakly connected, we can apply our algorithm on each component. Choose an arbitrary vertex $r \in V(G)$. Let L_0 be the set containing r and all vertices reachable from r by a directed path. Let $L_1 \overset{\text{def}}{=} \{v \in G \setminus L_0 : \exists u \in L_0 \text{ such that } v \rightsquigarrow u\}$. Similarly for $i \geq 1$, let $L_{2i} \overset{\text{def}}{=} \{v \in G \setminus \cup_{j=0}^{2i-1} L_j : \exists u \in \cup_{j=0}^{2i-1} L_j \text{ such that } u \rightsquigarrow v\}$ and $L_{2i+1} \overset{\text{def}}{=} \{v \in G \setminus \cup_{j=0}^{2i} L_j : \exists u \in \cup_{j=0}^{2i} L_j \text{ such that } v \rightsquigarrow u\}$. Then $L_0, L_1, \ldots, L_t$ partition the vertices of G, for some integer $t \leq n$. The following claim follows easily by the definitions of L_j.

CLAIM 4.4. *For all vertices $u, v \in G$, if $u \rightsquigarrow_G v$ and if $u \in L_i$ and $v \in L_j$, then $|i - j| \leq 1$.*

For $1 \leq i \leq t$, let $G_i \overset{\text{def}}{=} L_{i-1} \cup L_i$. By Claim 4.4, any two vertices with a dipath between them must both be contained in some G_i. Moreover, any dipath between them must lie entirely in G_i. Therefore, a k-TC-spanner of G is the union of k-TC-spanners of all G_i. Notice that $\sum_i |V(G_i)| \leq 2|V(G)|$.

We next construct a spanning tree T_G for G', the *undirected* graph underlying G, that is rooted at r and has the following property: for any monotone path in T_G from the root, the restriction of the path to a single level L_i consists of a single directed path in G.

T_G can be constructed inductively. First, since by definition r reaches all the vertices in L_0, a spanning tree of L_0 rooted at r can be constructed with all edges oriented away from r. Now, suppose we have a tree T_{i-1} that is rooted at r, spans all vertices in $\cup_{j=0}^{i-1} L_j$, and the restriction of any monotone path in T_{i-1} from the root to each level $0, \ldots, i-1$ consists of a single directed path. If i is odd, by the definition of L_i, T_{i-1} can be extended to a tree T_i so that T_i spans all vertices in $\cup_{j=0}^{i} L_j$ and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

all the new edges of T_i are oriented towards $\cup_{j=0}^{i-1} L_j$. Similarly, when i is even, by the definition of L_i, T_{i-1} can be extended to a tree T_i so that T_i spans all vertices in $\cup_{j=0}^{i} L_j$ and all the new edges of T_i are oriented from $\cup_{j=0}^{i-1} L_j$. Our desired spanning tree T_G is T_t. The following lemma is immediate by the construction.

LEMMA 4.5. *For all $i \in [t]$, a monotone path in T_G restricted to G_i is a concatenation of at most two dipaths.*

Case $k = 2$. We now describe how to construct H, a 2-TC-spanner of G.

The recursive graph fragmentation. First, we apply the preprocessing step described above to $G^0 \overset{\text{def}}{=} G$; that is, we obtain a spanning tree T_{G^0} and a collection of subgraphs $G_1^0, G_2^0, \dots, G_{\ell_0}^0$ for some $\ell_0 \leq n$. By definition of path separability, there exists a set P^0 of monotone paths on T_{G^0} such that one of the two situations occurs: (1) all the connected components in $G^0 \setminus P^0$ are of size at most $n/2$, (2) the largest component of $G^0 \setminus P^0$ is of size greater than $n/2$ and is path separable. If (2) holds, let G^1 denote the induced subgraph of G^0 on the largest component of $G^0 \setminus P^0$. We can apply the preprocessing to G^1 to obtain a collection of subgraphs $G_1^1, G_2^1, \dots, G_{\ell_1}^1$, for some $\ell_1 \leq n$ and a spanning tree T_{G^1} rooted at some arbitrary vertex in G^1. Again, we find an appropriate set of paths P^1 in T_{G^1}, and we recurse if necessary on the largest component of $G^1 \setminus P^1$. The recursion ends when the graph has been disconnected into components of size at most $n/2$. Notice that the total number of paths in $P^0 \cup P^1 \cup \cdots$ is at most $s = \Theta(1)$ since G is s-path separable, and we then recurse only a constant number of times. Let $S \overset{\text{def}}{=} P^0 \cup P^1 \cup \cdots$ be the set of all the paths in the path separator. By Theorem 4.2, S can be computed in polynomial time.

Connecting the cut pairs in G. Call a pair of vertices (u, v) a *cut pair* if $u \rightsquigarrow_G v$ and every directed path from u to v intersects a path in S. We show how to construct edges of H that connect every cut pair by a path of length at most 2.

Do the following for every vertex $v \in V(G)$. Let $I = \{i : v \in V(G^i)\}$ and, additionally, for each $i \in I$, let $J_i = \{j : v \in V(G_j^i)\}$. Note that $|J_i| \leq 2$ for all $i \in I$. For each $i \in I$ and each $j \in J_i$, let P_j^i denote the restriction of the paths in P^i to G_j^i. Each undirected path in P_j^i is a concatenation of at most two directed paths by Lemma 4.5. Break up the paths in P_j^i into at most two dipaths. Consider some dipath $P \in P_j^i$ which visits the vertices $p_1, p_2, \dots, p_m$ in that order, where $m \leq |V(G_j^i)|$. For simplicity of presentation, so as to not worry about floors and ceilings, assume that m is a power of 2 minus 1. First, choose the midpoint $p_{(m+1)/2}$ as a "hub." For all the vertex $v \in V(G_j^i)$ such that $v \rightsquigarrow_G p_{(m+1)/2}$, add an edge from v to $p_{(m+1)/2}$ to H. Similarly for all the vertex $v \in V(G_j^i)$ such that $p_{(m+1)/2} \rightsquigarrow_G v$, add to H an edge from $p_{(m+1)/2}$ to v. These edges connect all the cut pairs $(u, v) \in V(G_j^i) \times V(G_j^i)$ such that $\min_\ell\{1 \leq \ell \leq m : u \rightsquigarrow_G p_\ell\} \leq (m+1)/2$ and $\max_\ell\{1 \leq \ell \leq m : p_\ell \rightsquigarrow_G v\} \geq (m+1)/2$ by a path of length at most 2. These edges are called the edges of level 1. In the next level 2, we recursively choose two midpoints $p_{(m+1)/4}$ and $p_{3(m+1)/4}$ of the two halves as additional hubs. For each vertex $v \in V(G_j^i)$ such that $v \rightsquigarrow_G p_{3(m+1)/4}$, add to H an edge from v to $p_{y_1 \cdot (m+1)/4}$, where $y_1 = \min_y\{1 \leq y \leq 3 : v \rightsquigarrow_G p_{y \cdot (m+1)/4}\}$. Similarly for each vertex $v \in V(G_j^i)$ such that $p_{(m+1)/4} \rightsquigarrow_G v$, add an edge from $p_{y_2 \cdot (m+1)/4}$ to v where $y_2 = \max_y\{1 \leq y \leq 3 : p_{y \cdot (m+1)/4} \rightsquigarrow_G v\}$ to H. Then the new edges of H added in the level 2 connect all the cut pairs $(u, v) \in V(G_j^i) \times V(G_j^i)$ such that $\min_y\{1 \leq y \leq 3 : u \rightsquigarrow_G p_{y \cdot (m+1)/4}\} = \max_y\{1 \leq y \leq 3 : p_{y \cdot (m+1)/4} \rightsquigarrow_G v\}$ by a path of length at most 2. Repeat this process for every level. Formally, for each level $1 \leq z \leq \log_2(m+1)$, we add (at most) two edges in H as follows. (i) An edge from v

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

to $p_{y_1\cdot(m+1)/2^z}$, where $y_1 = \min_y\{1 \leq y \leq 2^z : v \rightsquigarrow p_{y\cdot(m+1)/2^z} \text{ in } G\}$ and (ii) an edge to v from $p_{y_2\cdot(m+1)/2^z}$, where $y_2 = \max_y\{1 \leq y \leq 2^z : p_{y\cdot(m+1)/2^z} \rightsquigarrow v \text{ in } G\}$. If any of the sets inside the min or max is empty, do not add the respective edge. Repeat this process for every separator dipath in S that is a subpath of an undirected path in some P_j^i.

Outer recursion. For each connected component C of $G \setminus S$, recurse the above process on the subgraph induced by C. Note that C is also path separable since the graph family is minor-closed.

LEMMA 4.6. *The above construction produces a 2-TC-spanner of G of size $O(n \log^2 n)$ in polynomial time.*

Proof. Let us first see why connecting every cut pair by a path of length at most 2 and recursing on smaller components produce a 2-TC-spanner of G. Indeed, if (u, v) is a cut pair, then the first step ensures a path of length at most 2 between them. If (u, v) is not a cut pair but there exist some dipaths from u to v, then u and v are in the same component C of $G \setminus S$, and there exists a dipath between them that lies entirely within this component. In this case, constructing a 2-TC-spanner of the subgraph induced by this C suffices to connect u and v by a path of length at most 2.

Let us now argue that our process connects every cut pair by a path of length at most 2. Consider some cut pair (u, v). Let i be the smallest nonnegative integer such that every dipath from u to v intersects a path in $\cup_{i' \leq i} P_{i'}$. Therefore, there must be a dipath from u to v entirely contained in G^i, and by Claim 4.4, it follows that there is a j such that both u and v are in G_j^i. Suppose $P \in P_j^i$ is a separator dipath of length m (a power of 2) that intersects a dipath in G^i from u to v, and as before, let $p_1, \ldots, p_m$ be the vertices on P in that order. Let $y_1 = \min_y\{y : u \rightsquigarrow_{G^i} p_y\}$ and $y_2 = \max_y\{y : p_y \rightsquigarrow_{G^i} v\}$. $y_1 \leq y_2$ because, otherwise, there cannot be a vertex on P that lies on a path from u to v. Then there exists some level $z \in \{1, 2, \ldots, \log_2(m+1)\}$ such that there is a unique $y \in \{1, 2, \ldots, 2^z - 1\}$ for which $y \cdot (m + 1)/2^z$ is in the interval $[y_1, y_2]$. Moreover, $u \rightsquigarrow p_{y_1} \rightsquigarrow p_{y\cdot(m+1)/2^z} \rightsquigarrow p_{y_2} \rightsquigarrow v$. Therefore, the edges $(u, p_{y\cdot(m+1)/2^z})$ and $(p_{y\cdot(m+1)/2^z}, v)$ are in H by our construction.

In connecting the cut pairs in G_j^i, we add $O(s\,|V(G_j^i)|\,\log|V(G_j^i)|)$ edges because there are $O(s)$ separator dipaths in G_j^i and for any separator dipath P, each vertex in G_j^i is connected to at most $2(\log_2|V(P)| + 1) = O(\log|V(G_j^i)|)$ vertices on the path. Recall that there are only a constant number of G^is and $\sum_j |V(G_j^i)| \leq 2|V(G^i)|$. Thus, if $S(G)$ denotes the total number of edges in the constructed 2-TC-spanner of G, then

$$\begin{aligned} S(G) &\leq \sum_{C \text{ is a c.c. of } G\setminus S} S(C) + O\left(\max_i \sum_j O(|V(G_j^i)| \cdot \log|V(G_j^i)|)\right) \\ &\leq \sum_{C \text{ is a c.c. of } G\setminus S} S(C) + O(n \log n). \end{aligned}$$

Since $|V(C)| \leq n/2$ for any connected component of $G \setminus S$, it follows that $S(G) = O(n \log^2 n)$.

Since the path separators can be found in polynomial time, as guaranteed in Theorem 4.2, it is clear that the above 2-TC-spanner can be constructed in polynomial time. ∎

Case $k > 2$. We now prove Theorem 4.3 for general k. As before, assume that G is transitively reduced and connected. We construct H, a k-TC-spanner of G.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We perform the same preprocessing as before in order to obtain induced subgraphs $G^0, G^1, \ldots$ and a corresponding s-path separator $S = P^0 \cup P^1 \cup \cdots$. Define a cut pair (u, v) to be a pair of vertices in G such that $u \rightsquigarrow v$ and every directed path from u to v intersects a path in S. This time, our plan is to connect all cut pairs by a path of length at most k and then to recurse on each of the connected components that remain after removing the vertices in the paths of S. By the argument used earlier, this process produces a k-TC-spanner in polynomial time.

Now we show how to connect cut pairs (u, v) with a path of length at most k. Do the following for every vertex $v \in V(G)$. Let $I = \{i : v \in V(G^i)\}$ and for each $i \in I$, let $J_i = \{j : v \in V(G^i_j)\}$. Do the following for each $i \in I$ and for each $j \in J_i$. Let P^i_j be the restriction of the paths in P^i to G^i_j. Break up the undirected paths into dipaths, increasing the size of P^i_j by a factor of at most 2. Do the following for each dipath $P \in P^i_j$. Let m be the length of P which visits vertices $p_1, p_2, \ldots p_m$ in that order. The construction is similar to the case $k = 2$, except that each vertex in G is connected to fewer vertices in S and, additionally, we add to H edges of a $(k-2)$-TC-spanner for subpaths of the dipath $\{p_1, p_2, \ldots, p_m\}$.

Let $c(\ell)$ be a concave increasing function of ℓ, such that $\ell/c(\ell)$ is an increasing function of ℓ (and hence, $c(\ell) < \ell$). We will exactly specify the function $c(\ell)$ later. For simplicity of presentation, we omit all floors and ceilings. Let $c^*(\ell)$ denote the smallest z such that $c^z(\ell) \le 3$, where $c^z(\cdot)$ denotes the zth functional power of c. As in the case $k = 2$, for each z such that $1 \le z \le c^*(m)$, for each vertex v of G, add the following two edges to H: (i) an edge from v to $p_{y_1 \cdot c^z(m)}$, where $y_1 = \min_y\{1 \le y \le m/c^z(m) : v \rightsquigarrow p_{y \cdot c^z(m)}$ in $G\}$ and (ii) an edge to v from $p_{y_2 \cdot c^z(m)}$, where $y_2 = \max_y\{1 \le y \le m/c^z(m) : p_{y \cdot c^z(m)} \rightsquigarrow v$ in $G\}$. If any of the sets inside the min or max is empty, do not add the respective edge.

Finally, do the following for every dipath $P \in P^i_j$ that visits vertices $p_1, p_2, \ldots, p_m$ in that order.

```
ALGORITHM. Connect-On (Input: dipath $P$).
    1. Add to $H$ the edges of a $(k-2)$-TC-spanner for the directed path on
       vertices $p_{c(m)}, p_{2c(m)}, \ldots, p_m$ in that order.
    2. Remove the vertices $\{p_{c(m)}, p_{2c(m)}, \ldots, p_m\}$ from $P$ and run Connect-On
       on each connected component of $P$ that remains.
```

Our construction of H is completed by recursing on components of size less than $n/2$ that remain in $G \setminus S$. It is not hard to see that H is indeed a k-TC-spanner, using the same argument as in the case $k = 2$. The only difference is that now, for a cut pair (u, v), it could be that u and v are adjacent to different vertices on the separating dipath. But we have the guarantee, by CONNECT-ON above, that two path vertices in the same recursion level of CONNECT-ON have a path of length at most $k - 2$ between them. Hence, it follows that u and v have a path of length at most k between them.

Now, we bound the size of H. Let us count the number of edges added in each step of the outer recursion (that is, before recursing on components of $G \setminus S$). For each vertex v and for each separating dipath P, we add $O(c^*(n)$ edges incident to v. Since the total number of paths is $O(1)$, the total number of edges added to connect vertices to separating dipaths is $O(nc^*(n))$. Next, we bound the number of edges added by the CONNECT-ON procedure. The goal is to show that this bound also is $O(nc^*(n))$. Denote by $\ell(n, k)$ the quantity $S_k(L_n)$, the size of the optimal k-TC-spanner of the directed path on n vertices. Let $f(m)$ be the number of edges added to H that are

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

incident to some particular separating dipath $P \in P^i_j$ of size m. Then

$$f(m) \leq \ell\left(\frac{m}{c(m)}, k-2\right) + \frac{m}{c(m)} \cdot f(c(m)) \leq \ell\left(\frac{n}{c(n)}, k-2\right) + \frac{n}{c(n)} \cdot f(c(n)),$$

where the first term comes from the first step of CONNECT-ON and the second term comes from the second step. In order for $f(m)$ to be $O(nc^*(n))$, it can be seen that we need to have $\ell(\frac{n}{c(n)}, k-2) = O(n)$. Using the fact from [5] that $\ell(n,k) = \Theta(n \cdot \lambda_k(n))$, we get $c^*(n) = O(\lambda_k(n))$. This choice makes the solution of the above recurrence $f(m) \leq O(nc^*(n)) = O(n\lambda_k(n))$. Finally, since there are $\log n$ levels of the recursion at the top level (at each level, the size of the largest component is decreased by a factor of 2), the total number of edges added to H is $O(n \cdot \log n \cdot c^*(n)) = O(n \cdot \log n \cdot \lambda_k(n))$, as desired. ◻

Acknowledgments. We would like to thank the anonymous referees for the useful comments and Michael Elkin, Vitaly Feldman, Cyril Gavoille, Elad Hazan, Piotr Indyk, T.S. Jayram, Ronitt Rubinfeld, and Adam Smith for helpful discussions.

REFERENCES

[1] I. Abraham and C. Gavoille, *Object location using path separators*, in Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 2006, pp. 188–197.

[2] A. V. Aho, M. R. Garey, and J. D. Ullman, *The transitive reduction of a directed graph*, SIAM J. Comput., 1 (1972), pp. 131–137.

[3] N. Ailon and B. Chazelle, *Information theory in property testing and monotonicity testing in higher dimension*, Inform. and Comput., 204 (2006), pp. 1704–1717.

[4] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani, *Fast estimation of diameter and shortest paths (without matrix multiplication)*, SIAM J. Comput., 28 (1999), pp. 1167–1181.

[5] N. Alon and B. Schieber, *Optimal Preprocessing for Answering On-line Product Queries*, Technical Report 71/87, Tel Aviv University, Tel Aviv, Israel, 1987.

[6] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, *On sparse spanners of weighted graphs*, Discrete Comput. Geom., 9 (1993), pp. 81–100.

[7] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid, *Euclidean spanners: Short, thin, and lanky*, in Proceedings of the 27th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1995, pp. 489–498.

[8] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, *Dynamic and efficient key management for access hierarchies*, ACM Trans. Inf. Syst. Secur., 12 (2009), pp. 1–43.

[9] M. J. Atallah, M. Blanton, and K. B. Frikken, *Key management for non-tree access hierarchies*, in Proceedings of the 11th Annual Symposium on Access Control Models and Technologies, ACM, New York, 2006, pp. 11–18.

[10] M. J. Atallah, K. B. Frikken, and M. Blanton, *Dynamic and efficient key management for access hierarchies.*, in Proceedings of the 12th ACM Conference on Computer and Communications Security, ACM, New York, 2005, pp. 190–202.

[11] B. Awerbuch, *Communication-time trade-offs in network synchronization*, in Proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 1985, pp. 272–276.

[12] S. Baswana and S. Sen, *Approximate distance oracles for unweighted graphs in expected $\tilde{O}(n^2)$ time*, ACM Trans. Algorithms, 2 (2006), pp. 557–577.

[13] P. Berman, A. Bhattacharyya, E. Grigorescu, S. Raskhodnikova, D. P. Woodruff, and G. Yaroslavtsev, *Steiner transitive-closure spanners of low-dimensional posets*, Lecture Notes in Comput. Sci. 6755, Springer, Heidelberg, 2011, pp. 760–772.

[14] P. Berman, A. Bhattacharyya, K. Makarychev, S. Raskhodnikova, and G. Yaroslavtsev, *Improved approximation for the directed spanner problem*, Lecture Notes in Comput. Sci. 6755, Springer, Heidelberg, 2011, pp. 1–12.

[15] P. Berman and M. Karpinski, *On Some Tighter Inapproximability Results*, Technical Report 85193-CS, DIMACS, Piscataway, NJ, 1998.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

[16] P. Berman, S. Raskhodnikova, and G. Ruan, *Finding sparser directed spanners*, in Proceedings of the 30th International Conference on Foundations of Software Technology and Theoretical Computer Science, LIPIcs Leibniz Int. Proc. Inform., Schloss Dagstuhl, Leibniz-Zent. Inform., Wadern, Germany, 2010, pp. 424–435.

[17] A. Bhattacharyya, E. Grigorescu, M. Jha, K. Jung, S. Raskhodnikova, and D. P. Woodruff, *Lower bounds for local monotonicity reconstruction from transitive-closure spanners*, SIAM J. Discrete Math., 26 (2012), pp. 618–646.

[18] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. Woodruff, *Transitive-closure spanners*, in Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2009, pp. 932–941.

[19] H. L. Bodlaender, G. Tel, and N. Santoro, *Tradeoffs in non-reversing diameter*, Nordic J. Comput., 1 (1994), pp. 111–134.

[20] A. K. Chandra, S. Fortune, and R. J. Lipton, *Lower bounds for constant depth circuits for prefix problems*, in ICALP, Josep Díaz, ed., Lecture Notes in Comput. Sci. 154, Springer, Heidelberg, 1983, pp. 109–117.

[21] A. K. Chandra, S. Fortune, and R. J. Lipton, *Unbounded fan-in circuits and associative functions*, J. Comput. System Sci., 30 (1985), pp. 222–234.

[22] B. Chazelle, *Computing on a free tree via complexity-preserving mappings*, Algorithmica, 2 (1987), pp. 337–361.

[23] C. Chekuri, G. Even, A. Gupta, and D. Segev, *Set connectivity problems in undirected graphs and the directed steiner network problem*, ACM Trans. Algorithms, 7 (2011), article 18.

[24] E. Cohen, *Fast algorithms for constructing t-spanners and paths with stretch t*, SIAM J. Comput., 28 (1999), pp. 210–236.

[25] E. Cohen, *Polylog-time and near-linear work approximation scheme for undirected shortest paths*, J. ACM, 47 (2000), pp. 132–166.

[26] L. J. Cowen, *Compact routing with minimum stretch*, J. Algorithms, 38 (2001), pp. 170–183.

[27] L. J. Cowen and C. G. Wagner, *Compact roundtrip routing in directed networks*, J. Algorithms, 50 (2004), pp. 79–95.

[28] M. Dinitz and R. Krauthgamer, *Directed spanners via flow-based linear programs*, in Proceedings of the 43rd Annual ACM Symposium on the Theory of Computing, ACM, New York, 2011, pp. 323–332.

[29] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky, *Improved testing algorithms for monotonicity*, in RANDOM, Lecture Notes in Comput. Sci. 1671, Springer, Berlin, 1999, pp. 97–108.

[30] Y. Dodis and S. Khanna, *Designing networks with bounded pairwise distance*, in Proceedings of the 31st Annual ACM Symposium on the Theory of Computing, ACM, New York, 1999, pp. 750–759.

[31] D. Dor, S. Halperin, and U. Zwick, *All-pairs almost shortest paths*, SIAM J. Comput., 29 (2000), pp. 1740–1759.

[32] M. Elkin, *Computing almost shortest paths*, ACM Trans. Algorithms, 1, 2005, pp. 283–323.

[33] M. Elkin and D. Peleg, *Strong inapproximability of the basic k-spanner problem*, in Proceedings of the 27th Annual International Conference on Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 1853, Springer, Berlin, 2000, pp. 636–647.

[34] M. Elkin and D. Peleg, *The client-server* 2*-spanner problem with applications to network design*, in Proceedings of the 8th International Colloquium on Structural Information and Communication Complexity, Carleton Scientific, Ontario, Canada, 2001, pp. 117–132.

[35] M. Elkin and D. Peleg, *Approximating k-spanner problems for $k \geq 2$*, Theoret. Comput. Sci., 337 (2005), pp. 249–277.

[36] M. Elkin and D. Peleg, *The hardness of approximating spanner problems*, Theory Comput. Syst., 41 (2007), pp. 691–729.

[37] D. Eppstein, *Finding the k shortest paths*, SIAM J. Comput., 28 (1998), pp. 652–673.

[38] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan, *Spot-checkers*, J. Comput. System Sci., 60 (2000), pp. 717–751.

[39] U. Feige, *A threshold of $\ln n$ for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[40] M. Feldman, G. Kortsarz, and Z. Nutov, *Improved approximating algorithms for Directed Steiner Forest*, in Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2009, pp. 922–931.

[41] E. Fischer, *On the strength of comparisons in property testing*, Inform. and Comput., 189 (2004), pp. 107–116.

[42] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky, *Monotonicity testing over general poset domains*, in Proceedings of the 34th Annual ACM Symposium on the Theory of Computing, ACM, New York, 2002, pp. 474–483.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[43] C. Gavoille, *Personal communication*, 2007.
[44] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky, *Testing monotonicity*, Combinatorica, 20 (2000), pp. 301–337.
[45] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. ACM, 45 (1998), pp. 653–750.
[46] S. Halevy and E. Kushilevitz, *Testing monotonicity over graph products*, in Proceedings of the 31st Annual International Conference on Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 3142, Springer, Berlin, 2004, pp. 721–732.
[47] W. Hesse, *Directed graphs requiring large numbers of shortcuts*, in Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2003, pp. 665–669.
[48] D. Hochbaum, ed., *Approximation Algorithms for NP-hard Problems*, PWS Publishing Company, Boston, 1997.
[49] M. Jha and S. Raskhodnikova, *Testing and reconstruction of Lipschitz functions with applications to data privacy*, in Proceedings of the FOCS, 2011, pp. 433–442.
[50] G. Kortsarz, *On the hardness of approximating spanners*, Algorithmica, 30 (2001), pp. 432–450.
[51] G. Kortsarz and D. Peleg, *Generating sparse* 2*-spanners*, J. Algorithms, 17 (1994), pp. 222–236.
[52] G. Kortsarz and D. Peleg, *Generating low-degree* 2*-spanners*, SIAM J. Comput., 27 (1998), pp. 1438–1456.
[53] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.
[54] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, in Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1993, pp. 286–293.
[55] W. Mader, *Homomorphieeigenschaften und mittlere kantendichte von graphen*, Math. Ann., 174 (1967), pp. 265–268.
[56] G. Narasimhan and M. H. M. Smid, *Geometric Spanner Networks*, Cambridge University Press, Cambridge, UK, 2007.
[57] D. Peleg, *Distributed Computing: A Locality-sensitive Approach*, SIAM, Philadelphia, 2000.
[58] D. Peleg and A. A. Schäffer, *Graph spanners*, J. Graph Theory, 13 (1989), pp. 99–116.
[59] D. Peleg and J. D. Ullman, *An optimal synchronizer for the hypercube*, SIAM J. Comput., 18 (1989), pp. 740–747.
[60] D. Peleg and E. Upfal, *A trade-off between space and efficiency for routing tables*, J. ACM, 36 (1989), pp. 510–530.
[61] S. Raskhodnikova, *Transitive-closure spanners: A survey*, in Property Testing, Oded Goldreich, ed., Lecture Notes in Comput. Sci. 6390, Springer, Heidelberg, 2010, pp. 167–196.
[62] R. Raz and S. Safra, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, in Proceedings of the 29th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1997, pp. 475–484.
[63] L. Roditty, M. Thorup, and U. Zwick, *Roundtrip spanners and roundtrip routing in directed graphs*, in Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2002, pp. 844–851.
[64] R. Rubinfeld and M. Sudan, *Robust characterization of polynomials with applications to program testing*, SIAM J. Comput., 25 (1996), pp. 252–271.
[65] A. De Santis, A. L. Ferrara, and B. Masucci, *Efficient provably-secure hierarchical key assignment schemes*, in Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Comput. Sci. 4708, Springer, Berlin, 2007, pp. 371–382.
[66] S. Solomon, *An optimal-time construction of sparse Euclidean spanners with tiny diameter*, in Proceedings of 22nd ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2011, pp. 820–839.
[67] S. Solomon and M. Elkin, *Balancing degree, diameter and weight in Euclidean spanners*, in Proceedings of the 18th ESA, Lecture Notes in Comput. Sci. 6346, Springer, Berlin, 2010, pp. 48–59.
[68] M. Thorup, *On shortcutting digraphs*, E. W. Mayr, ed., Lecture Notes in Comput. Sci. 657, Springer, Berlin, 1993, pp. 205–211.
[69] M. Thorup, *Shortcutting planar digraphs*, Combin. Probab. Comput., 4 (1995), pp. 287–315.
[70] M. Thorup, *Parallel shortcutting of rooted trees*, J. Algorithms, 23 (1997), pp. 139–159.
[71] M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*, J. ACM, 51 (2004), pp. 993–1024.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[72] M. Thorup and U. Zwick, *Compact routing schemes*, in Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures, ACM, New York, 2001, pp. 1–10.

[73] M. Thorup and U. Zwick, *Approximate distance oracles*, J. ACM, 52 (2005), pp. 1–24.

[74] D. P. Woodruff, *Lower bounds for additive spanners, emulators, and more*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 2006, pp. 389–398.

[75] A. C.-C. Yao, *Space-time tradeoff for answering range queries (extended abstract)*, in Proceedings of the 14th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1982, pp. 128–136.

[76] U. Zwick, *Exact and approximate distances in graphs—A survey*, Lecture Notes in Comput. Sci. 2161, Springer, Berlin, 2001, pp. 33–48.

Downloaded 09/26/13 to 147.46.115.202. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.