Finding Sparser Directed Spanners*

Piotr Berman[1], Sofya Raskhodnikova[1], and Ge Ruan[1]

1 **Pennsylvania State University, USA**
{berman,sofya,gur112}@cse.psu.edu

Abstract

A spanner of a graph is a sparse subgraph that approximately preserves distances in the original graph. More precisely, a subgraph $H = (V, E_H)$ is a *k-spanner* of a graph $G = (V, E)$ if for every pair of vertices $u, v \in V$, the shortest path distance $dist_H(u, v)$ from u to v in H is at most $k \cdot dist_G(u, v)$. We focus on spanners of *directed graphs* and a related notion of *transitive-closure spanners*. The latter captures the idea that a spanner should have a small diameter but preserve the connectivity of the original graph. We study the computational problem of finding the sparsest k-spanner (resp., k-TC-spanner) of a given *directed* graph, which we refer to as Directed k-Spanner (resp., k-TC-Spanner). We improve all known approximation algorithms for Directed k-Spanner and k-TC-Spanner for $k \geq 3$. (For $k = 2$, the current ratios are tight, assuming P≠NP.) Along the way, we prove several structural results about the size of the sparsest spanners of directed graphs.

Keywords and phrases Approximation algorithms, directed graphs, spanners

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2010.424

1 Introduction

A spanner of a graph is a sparse subgraph that approximately preserves distances in the original graph.

▶ **Definition 1.1** (k-spanner, [3, 24])**.** A subgraph $H = (V, E_H)$ is a **k-spanner** of a graph $G = (V, E)$ if for all vertices $u, v \in V$, the shortest path distance $dist_H(u, v)$ from u to v in H is at most $k \cdot dist_G(u, v)$. The parameter k is called the **stretch**.

Graph spanners have numerous applications, ranging from efficient routing [13, 14, 26, 31] and simulating synchronized protocols in unsynchronized networks [25] to parallel and distributed algorithms for approximating shortest paths [11, 12, 15] and algorithms for distance oracles [5, 32].

We focus on spanners of *directed graphs* and a related notion of *transitive-closure spanners* studied in [7, 28, 29, 30, 19, 8, 6]. The latter captures the idea that a spanner should have a small diameter but preserve the connectivity of the original graph. The diameter here means the largest distance between a pair (u, v) of nodes in a directed graph such that v is reachable from u. Recall that the *transitive closure* of a graph $G = (V, E)$, denoted $TC(G)$, is a graph (V, E_{TC}) where $(u, v) \in E_{TC}$ if and only if G has a directed path from u to v.

▶ **Definition 1.2** (k-TC-spanner, [7])**.** A directed graph $H = (V, E_H)$ is a **k-transitive-closure-spanner** (**k-TC-spanner**) of a directed graph $G = (V, E)$ if

1. E_H is a subset of the edges in the transitive closure of G and

* S.R. was supported by National Science Foundation (NSF/CCF award 0729171 and NSF/CCF CAREER award 0845701).

2. for all vertices $u, v \in V$, if $dist_G(u, v) < \infty$, then $dist_H(u, v) \leq k$.

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, such as managing keys in access control hierarchies, data structures for computing partial products in a semigroup, property testing and property reconstruction (see the survey in [27] and references therein).

In this paper, we study the computational problem of finding the sparsest k-TC-spanner (resp., k-spanner) of a given *directed* graph, which we refer to as k-TC-Spanner (resp., Directed k-Spanner).

1.1 Previous Work

To put the following results in proper context, observe that if a graph G has n vertices, every k-spanner of G has $O(n^2)$ edges.

1.1.1 Computational Results

All algorithms for Directed k-Spanner immediately yield algorithms for k-TC-Spanner with the same approximation ratio because k-TC-Spanner on input graph G is equivalent to Directed k-Spanner on input TC(G). Elkin and Peleg [17] gave an $O(\log n)$-approximation algorithm for Directed 2-Spanner. Kortsarz [22] showed that the $O(\log n)$ approximation ratio for Directed 2-Spanner cannot be improved unless P=NP, and [7] extended this result to 2-TC-spanner. Thus, for $k = 2$ the ratio for both problems has been completely resolved.

For $k = 3$, the first non-trivial approximation ratio for Directed k-Spanner was proved by Elkin and Peleg [16] and slightly improved (by a factor polylogarithmic in n) by Bhattacharyya *et al.* [7]. In general, for $k \geq 3$, [7] gives an approximation ratio $O((n \log n)^{1-1/k})$. For all $\delta, \epsilon \in (0, 1)$ and $3 \leq k \leq n^{1-\delta}$, it is impossible to approximate Directed k-Spanner within a factor of $2^{\log^{1-\epsilon} n}$ in polynomial time, assuming NP$\not\subseteq$DTIME($n^{\text{polylog}\, n}$) [16, 18]. ($DTIME(f(n))$ denotes the class of languages decidable deterministically in time $f(n)$.) Thus, according to Arora and Lund's classification [20] of NP-hard problems, Directed k-Spanner is in class III, for $k \in [3, n^{1-\delta}]$. [7] extended this result to k-TC-Spanner for constant k. [18] also 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 long standing open question in complexity theory: namely, cause classes III and IV to collapse into a single class. Since k-TC-Spanner is a special case of Directed k-Spanner, this statement also applies to the former.

For the special case of k-TC-Spanner, [7] presented an $O\left(\frac{n \log n}{k^2 + k \log n}\right)$-approximation algorithm. Observe that for large k, this is better than the inapproximability of Directed k-Spanner, demonstrating that k-TC-Spanner is a strictly easier problem for this range of parameters.

1.1.2 Structural Results

Unlike in the undirected setting, where for every $k \geq 1$, all graphs on n vertices have $(2k-1)$-spanners with $O(n^{1+1/k})$ edges [2, 23, 32], sparsest TC-spanners (and hence, sparsest directed spanners) can have $\Omega(n^2)$ edges. An example of a graph with a sparsest 2-TC-spanner of

Problem	Stretch k	Hardness	Known Ratio	Our Ratio
Directed spanner and TC-spanner	2	$\Omega(\log n)$ [22, 7]	$O(\log n)$ [17]	–
	3	$\Omega(2^{\log^{1-\epsilon} n})$ [16, 18, 7]	$O((n \log n)^{2/3})$ [16, 7]	$O(\sqrt{n} \log n)$
Directed Spanner	> 3	as above, for $k \leq n^{1-\delta}$ [16, 18]	$O((n \log n)^{1-1/k})$ [7]	$O(k \cdot n^{1-1/\lceil k/2 \rceil} \log n)$
TC-spanner	> 3	as above, for constant k [7]	as above	$O(n^{1-1/\lceil k/2 \rceil} \log n)$
	$\Omega\left(\frac{\log n}{\log\log n}\right)$	$(1+\delta)$, for $k \leq n^{1-\delta}$ [7]	$O\left(\frac{n \log n}{k^2 + k \log n}\right)$ [7]	$O\left(\frac{n}{k^2}\right)$

Table 1 Summary of Results on Approximability of k-TC-Spanner and Directed k-Spanner

size $\Omega(n^2)$ is the complete bipartite graph $K_{\frac{n}{2},\frac{n}{2}}$ with $n/2$ vertices in each part and all edges directed from the first part to the second. Therefore, to the best of our knowledge, there are no combinatorial results about the sizes of sparsest spanners for general directed graphs.

There are many combinatorial results on the sizes of the sparsest k-TC-spanners but, for the reason mentioned above, almost all of them apply exclusively to special families of graphs: directed lines and trees [10, 30], planar graphs [28], H-minor-free graphs [7], multi-dimensional grids [6] and low-dimensional posets [9]. There are only two results that can be viewed as structural results for general graphs. The first one is the TC-spanner-specific algorithm from [7], which is obtained by showing that every graph can be turned into a k-TC-spanner by adding $O\left(\frac{n^2 \log n}{k^2 + k \log n}\right)$ shortcut edges. The second one is a construction of Hesse [19] giving a family of graphs with large TC-spanners and disproving Thorup's conjecture [28] that all directed graphs G on n nodes have TC-spanners with stretch polylogarithmic in n and size at most $2|G|$. (We use $|G|$ to denoted the number of edges in G.) For all small $\epsilon > 0$, Hesse constructed a family of graphs with $n^{1+\epsilon}$ edges for which all n^{ϵ}-TC-spanners require $\Omega(n^{2-\epsilon})$ edges.

1.2 Our Results and Techniques

1.2.1 Computational Results

Table 1 summarizes the best known results on approximability of k-TC-Spanner and Directed k-Spanner, alongside with our results on these problems. We improve all known approximation algorithms for Directed k-Spanner and k-TC-Spanner for $k \geq 3$. (As mentioned before, for $k = 2$, the current ratios are tight, assuming P$\neq$NP.)

First, we obtain $O(k \cdot n^{1-1/\lceil k/2 \rceil} \log n)$-approximation for Directed k-Spanner. In particular, for both Directed k-Spanner and k-TC-Spanner, the best previously known ratios were $O((n \log n)^{2/3})$ for $k = 3$ and $O((n \log n)^{3/4})$ for $k = 4$. Our algorithm improves both of these ratios to $O(\sqrt{n} \log n)$. Second, we give a slightly better $O(n^{1-1/\lceil k/2 \rceil} \log n)$-approximation for k-TC-Spanner. Finally, we also present a $O(n/k^2)$-approximation algorithm for k-TC-Spanner, which outperforms our first algorithm for $k = \Omega(\log n / \log\log n)$.

1.2.2 Our Techniques and Structural Results

The approach we take is very different from previous work. The best previously known algorithm for small $k > 2$ from [7] (that works for k-TC-Spanner and Directed k-Spanner) is based on solving a linear program and combining the solution with paths obtained by random sampling. Our approximation algorithm for k-TC-Spanner for small k is based on the following idea: since known approximation algorithms do much better for stretch $k = 2$, we use a 2-TC-spanner produced by the approximation algorithm for 2-TC-spanner to approximate the sparsest k-TC-spanner. To show that it provides a good approximation, we prove that a k-TC-spanner of G cannot be much sparser than a 2-TC-spanner of G. More precisely, we show how to transform a k-TC-spanner into a 2-TC-spanner while increasing the number of edges by a factor of $O(n^{1-1/\lceil k/2 \rceil})$. These results are presented in Section 2.

In Section 3, we present our algorithm for Directed k-Spanner. First, we note that the idea used to approximate k-TC-Spanner cannot be applied directly: as shown in Lemma 3.2, there are directed graphs where the ratio between the sizes of the sparsest $(k-1)$-spanner and k-spanner is $\Omega(n)$. Instead, we build on the ideas of [21, 17] which use star graphs to give an approximation algorithm for Directed 2-spanner. We generalize the notion of a star to k-stars. Recall that stars contain a central node s and nodes of distance 1 from s. Intuitively, k-stars also have a central node s and nodes of distance at most k from s. We show that a k-spanner of a directed graph G can be approximated well by a collection of k-stars such that each edge (u, v) of G is *covered* by some k-star in the collection—namely, that k-star contains a path of length at most k from u to v. Then we use a greedy algorithm to find such a collection of k-stars with nearly minimum cost, where the cost is the total number of edges in all k-stars in the collection.

Our algorithm for Directed k-Spanner also works for stretch $k = 2$. Even though an algorithm with the same (optimal) approximation ratio has already been presented by Elkin and Peleg [17], our algorithm gives a unified treatment for all small k, including 2, and is slightly simpler for the case of $k = 2$ than the algorithm in [17]. (Elkin and Peleg use a slightly different greedy method to find stars. They categorize edges already covered by selected stars into different types and calculate two types of density, φ-density and ρ-density, to evaluate the benefit of adding a new star to the partial 2-spanner.)

In Section 4, we show that our transformation from a k-TC-spannner to a 2-TC-spanner cannot be improved significantly. We show that this transformation is tight for $k = 3$ and $k = 4$ by giving a simple example. For $k \geq 5$, we prove that our transformation is nearly tight by extending Hesse's construction [19], which was originally devised to disprove Thorup's conjecture, as described above. For every sufficiently small positive ϵ, we exhibit graphs with $2k$-TC-spanners of size $n^{1+1/k}$ and no 2-TC-spanners of size less than $n^{2-\epsilon}$.

Our approximation algorithm for k-TC-Spanner for large k, presented in Section 5, is also based on a structural result. Namely, we show how to (efficiently) obtain a k-TC-spanner of any graph by adding $O(n^2/k^2)$ shortcut edges. This is based on a simple greedy procedure. The algorithm for large stretch in [7] was based on random sampling.

1.3 Preliminaries

For a directed graph G, we denote the number of edges in G by $|G|$ and the size of the sparsest k-TC-spanner of G by $S_k(G)$. (The size refers to the number of edges.) We say two nodes in G are *comparable* if one of these nodes is reachable from the other.

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* 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.

A *transitive reduction* of G is a digraph G' with the fewest edges for which $TC(G') = TC(G)$. As shown by Aho *et al.* [1], a transitive reduction of a given graph can be computed efficiently via a greedy algorithm. 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 edges in H that do not change its transitive closure, and finally uncontracts $v(C)$ to an arbitrary directed cycle on vertices in C, choosing a representative vertex of C to be incident to the edges incident to $v(C)$ in H'. Directed acyclic graphs have a unique transitive reduction. We say G is *transitively reduced* if G is equal to its own transitive reduction.

2 Approximation Algorithm for k-TC-Spanner for small k

This section presents an approximation algorithm for k-TC-Spanner that provides the best known ratio for all stretches $k = O(\log n / \log\log n)$.

▶ **Theorem 2.1.** *k-TC-Spanner can be approximated with ratio $O(n^{1-1/\lceil k/2\rceil}\log n)$ in polynomial time.*

The proof of this theorem appears at the end of this section. It relies on the following lemma and corollary that relate $S_k(G)$ and $S_2(G)$.

▶ **Lemma 2.2.** *Let G be a directed graph of diameter at most k. Then one can efficiently construct a 2-TC-spanner of G while increasing the number of edges by a factor of $O(n^{1-1/\lceil k/2\rceil})$.*

Proof. Let G be a directed graph on n nodes of diameter at most k. Set $d = n^{1/\lceil k/2\rceil}$. We call a vertex in G *high-degree* if its degree is at least d, and *low-degree* otherwise. (The degree of a node counts both incoming and outgoing edges.)

Transformation from k-TC-spanners to 2-TC-spanners
Input: directed graph $G = (V, E)$ of diameter k
1. Form graph H from G by adding shortcut edges as follows:
   a. Connect every *high-degree* vertex u to all vertices comparable to u.
   b. Consider the induced subgraph G' of G containing only *low-degree* vertices. Connect every vertex u in G' to all vertices that are reachable from u by a path of length at most $\lceil k/2\rceil$ in G'.
2. Output H.

To see that the resulting graph H is a 2-TC-spanner, consider a pair of vertices (u, v) with v reachable from u in G. Let P be a shortest path from u to v in G. If P contains a *high-degree* vertex w then in step 1(a) of the transformation, u was connected to v by a path of length at most 2 via w. Otherwise, all nodes in P are *low-degree*. Since G has diameter at most k, path P contains a node w such that $dist_G(u, w) \le \lceil k/2\rceil$ and $dist_G(w, v) \le \lceil k/2\rceil$. Then in step 1(b) of the transformation, u was connected to v by a path of length at most 2 via w. Thus, every comparable pair of nodes in H is connected by a path of length 2 or 1.

Since the number of edges in a graph is 1/2 of the sum of its vertex degrees, to prove the stated bound on $|H|/|G|$, it is enough to show that the degree of each vertex increases by a factor of $O(n^{1-1/\lceil k/2\rceil})$ when H is constructed from G. This holds for each *high-degree* node

because its degree increases from at least $d = n^{1/\lceil k/2 \rceil}$ to at most $2(n-1)$ (counting both incoming and outgoing edges). Each *low-degree* vertex v of degree $deg(v)$ connects to at most $deg(v) \cdot d$ vertices at distance 2, $deg(v) \cdot d^2$ vertices at distance 3, ..., $deg(v) \cdot d^{\lceil k/2 \rceil - 1}$ vertices at distance $\lceil k/2 \rceil$. So, the degree of v in H is at most $deg(v) \cdot (d^{\lceil k/2 \rceil} - 1)/(d-1)$, thus increasing by a factor of $O(d^{\lceil k/2 \rceil - 1}) = O(n^{(\lceil k/2 \rceil - 1)/\lceil k/2 \rceil}) = O(n^{1-1/\lceil k/2 \rceil})$, as desired. ◀

We obtain the following corollary by letting G in Lemma 2.2 be the sparsest k-TC-spanner of a given graph.

▶ **Corollary 2.3.** $\frac{S_2(G)}{S_k(G)} = O(n^{1-1/\lceil k/2 \rceil})$ *for any digraph* G. *In particular,* $\frac{S_2(G)}{S_3(G)} = O(\sqrt{n})$.

Proof of Theorem 2.1. Elkin and Peleg [17] gave an $O(\log n)$-approximation algorithm for Directed 2-spanner. Running this algorithm on the transitive closure of an input graph G will yield an $O(\log n)$-approximation to the sparsest 2-TC-spanner of G, which, by Corollary 2.3, is a $O(n^{1-1/\lceil k/2 \rceil} \log n)$-approximation to the sparsest k-TC-spanner of G. ◀

3 Approximation Algorithm for Directed k-Spanner

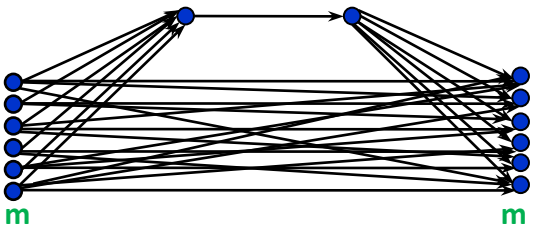
In this section, we present our approximation algorithm for Directed k-Spanner.

▶ **Theorem 3.1.** Directed k-Spanner *can be approximated with ratio* $O(k \cdot n^{1-1/\lceil k/2 \rceil} \log n)$ *in polynomial time.*

The following lemma demonstrates that the idea of approximating a k-spanner with a 2-spanner does not work for Directed k-Spanner.

▶ **Lemma 3.2.** *There is an infinite family of directed graphs* G *on* n *nodes for which every 2-spanner has* $\Omega(n^2)$ *edges while there is a* k*-spanner with* $O(n)$ *edges.*

Proof. Let m be a parameter such that $n = 2m + 2$. Our graph G has four layers of nodes, L_1, L_2, L_3 and L_4, of sizes $m, 1, 1$, and m, respectively. All edges are directed from smaller to larger layers. The following pairs of layers form complete bipartite graphs: (L_1, L_2), (L_2, L_3), (L_3, L_4) and (L_1, L_4). This completes the description of G.



G is the sparsest 2-spanner itself. It has $m^2 + 2m + 1 = \Omega(n^2)$ edges. To form a 3-spanner of G, we can delete all edges between layers L_1 and L_4. The resulting 3-spanner has $n - 1 = O(n)$ edges. By definition, a 3-spanner is also a k-spanner for all $k \geq 3$. ◀

Thus, approximating the sparsest k-spanner with a sparse 2-spanner will give an $\Omega(n)$ ratio in the worst case. Instead, we build on the ideas of [21, 17] which use star graphs to give an approximation algorithm for Directed 2-spanner.

▶ **Definition 3.3** (ℓ-Star, full ℓ-Star)**.** For a digraph G, a *1-star* is a complete 3-layered subgraph of G where all edges are directed from smaller to larger layers and the 2nd layer consists of a single node v. (Layers 1 and 3 can be empty.) More generally, an *ℓ-star* S_v is a $(2\ell + 1)$-layered subgraph of G where all edges start at some layer and end at the next

layer, and the middle layer consists of a single node v, called the *center*. Moreover, nodes in layers 1 to $\ell+1$ and edges between them form a shortest path tree (with respect to G) with root v (and edges directed towards the root). Similarly, nodes in layers $\ell+1$ to $2\ell+1$ and edges between them form a shortest path tree with root v (and edges directed away from the root). Some layers, can be empty, as long as the shortest path trees are valid. An ℓ-star that includes all vertices in G at distance at most ℓ from (to) the center is called *full*.

A full ℓ-star with center v can be found by performing two breadth-first searches on G starting from v (one on outgoing edges, another on incoming edges) and terminating both when all nodes at distance at most ℓ from (to) v have been found. Note that ℓ'-star is also an ℓ-star if $\ell' \leq \ell$. When ℓ is not specified or clear from the context, we omit it and call the corresponding graph a *star*.

In the context of finding directed k-spanners, we say that a star S *covers* an edge (u, v) of G if S contains a path of length at most k from u to v. A *covering* set C of stars is a set of stars such that each edge in G is covered by at least one of the stars in C. The *cost* of a set C of stars, denoted $cost(C)$, is the sum of the sizes of stars in the set. Observe that by taking all edges in the stars of a covering set C of stars, we obtain a directed k-spanner of size at most $cost(C)$.

Given a directed 2-spanner H of G, it is easy to construct a set C of 1-stars that cover every edge in G with $cost(C) \leq 2|H|$. E.g., we can include the full 1-star S_v with respect to H for each node in H. Since H is a 2-spanner of G, each edge (u, v) in G has a path of length at most 2 in H. This path is included in at least one of the stars in our set. Therefore, each edge is covered by one of the stars in the set. Since each edge is included in at most 2 stars, $cost(C) \leq 2|H|$.

The main idea of our algorithm for Directed k-Spanner is to cover all edges of G with stars. The next lemma shows that a covering set of k-stars with relatively low cost always exists.

▶ **Lemma 3.4.** *Let G be a directed graph on n nodes, and H be a k-spanner of G. Then there is a set C of k-stars that cover all edges in G with $cost(C) = O(n^{1-1/\lceil k/2 \rceil}|H|)$.*

Proof. We build on ideas in the proof of Lemma 2.2. As before, set $d = n^{1/\lceil k/2 \rceil}$. We call a vertex *high-degree* if its degree in H is at least d, and *low-degree* otherwise.

Transformation from a k-spanner to a covering set of stars

Input: a k-spanner H of G

1. Form a collection C of stars as follows:
   a. For every *high-degree* vertex u, add a full (with respect to H) k-star with center u.
   b. Consider the induced subgraph H' of H containing only *low-degree* vertices. For every vertex u in H', add a $\lceil k/2 \rceil$-star with center u containing all vertices v with $dist_{H'}(u, v)$ or $dist_{H'}(v, u)$ at most $\lceil k/2 \rceil$.
2. Output C.

First, we show that every edge (u, v) in G is covered by a star in C. Let P be a shortest path (of length at most k) from u to v in H. If P contains a *high-degree* vertex w then in step 1(a) of the transformation, a full k-star S_w with center w was added to C. By definition, S_w must contain both u and v as well as shortest paths from u to w and from w to v. Since w is on a shortest path u to v, it also contains a shortest path from u to v, and thus covers (u, v).

Otherwise, all nodes on path P are *low-degree*. Let w be the "middle" node on path P: that is, a node such that $dist_P(u,w) \leq \lceil k/2 \rceil$ and $dist_P(w,v) \leq \lceil k/2 \rceil$. Then in step 1(b) of the transformation, set C got a $\lceil k/2 \rceil$-star S_w with center w containing all nodes on path P. Since w is on a shortest path from u to v, this $\lceil k/2 \rceil$-star contains a shortest path from u to v via w. Thus, every edge in G is covered by a star in C.

Now we analyze $cost(C)$. Let $deg(u)$ denote the degree of node u in H. We add one star S_u to H for each node u. If u is a *high-degree* node, S_u has at most $2n = O(n^{1-1/\lceil k/2 \rceil}) \cdot deg(u)$ edges. Each *low-degree* vertex u connects to at most $deg(u)$ vertices at distance 1, $deg(u) \cdot d$ vertices at distance 2, $deg(u) \cdot d^2$ vertices at distance 3, ..., $deg(u) \cdot d^{\lceil k/2 \rceil - 1}$ vertices at distance $\lceil k/2 \rceil$. So, the number of edges in S_u is $O(d^{\lceil k/2 \rceil - 1}) \cdot deg(v) = O(n^{1-1/\lceil k/2 \rceil}) \cdot deg(u)$. Therefore, $cost(C) = \sum_{v \text{ in } H} |S_v| = O(n^{1-1/\lceil k/2 \rceil}) \cdot \sum_{v \text{ in } H} deg(u) = O(n^{1-1/\lceil k/2 \rceil}|H|)$. ◀

Next we prove the main theorem of this section.

Proof of Theorem 3.1. Our algorithm for Directed k-Spanner works by greedily constructing a covering set C of stars for the input graph G. As we mentioned, such a set easily yields a directed k-spanner of G of size $cost(C)$. Observe that the problem of finding a covering set of stars of minimum cost is a special case of Weighted Set Cover, where edges of G correspond to elements to be covered, and each star with $|S|$ edges corresponds to a covering set of weight (or cost) $|S|$. There are too many possible stars to write down the instance of Weighted Set Cover. However, recall that a simple greedy algorithm that chooses the set with the smallest relative cost (*i.e.,* the ratio of the cost of the set over the number of new elements it covers) gives approximation ratio $O(\log n)$ to Weighted Set Cover [33]. Therefore, if we had a subroutine that finds a star with the smallest relative cost in polynomial time, we would have a $O(\log n)$-approximation to our star covering problem. Instead, we will give a subroutine that finds a k-approximation to a star with the smallest relative cost, that is, it finds a star with relative cost at most k times the optimum. Then, the standard analysis of the greedy algorithm for Weighted Set Cover yields approximation ratio $O(k \log n)$.

It is not hard to see that a 1-star S_v with center v with smallest relative cost can be found in polynomial time using flow techniques (e.g., Project Selection). By finding such a star S_v for each v in G and selecting the best star, we can find the star with smallest relative cost. Incidentally, together with Lemma 3.4 and the observations above about Weighted Set Cover, this procedure gives a $O(\log n)$ approximation algorithm for Directed 2-Spanner.

Now let *opt* be the smallest relative cost of a k-star with center v. (Recall that an ℓ-star is also a k-star for $\ell \leq k$.) We will show how to find an k-star S_v with center v and relative cost at most $k \cdot opt$. Let G' be a graph obtained from G by adding edges from v (resp., to v) to all nodes reachable from v (resp., from all nodes from which v is reachable) at distance at most k. We run the subroutine above for finding a 1-star with center v of smallest relative cost on G'. It is guaranteed to output a 1-star of relative cost at most *opt*. This 1-star corresponds to a k-star in G of relative cost at most $k \cdot opt$, since the cost of adding each node is at most k instead of 1. As before, we run this subroutine for all nodes v and select the best star to obtain a k-approximation to a k-star with smallest relative cost.

We use the above subroutine to select k-stars for our collection C until all edges of G are covered. Let OPT be the smallest cost of a covering set of k-stars. As we explained, our algorithm will produce a covering set C of stars of cost $O(OPT \cdot k \log n)$. Let H be a sparsest k-spanner of G. By Lemma 3.4, there is a covering set C^* of k-stars with $cost(C^*) = O(n^{1-1/\lceil k/2 \rceil}|H|)$. Therefore, $cost(C) = O(cost(C^*) \cdot k \log n) = O(k \cdot n^{1-1/\lceil k/2 \rceil}(\log n)|H|)$, as required. ◀

4 Tightness of k-TC-spanner to 2-TC-spanner Transformation

In this section, we show that our transformation from a k-TC-spanner to a 2-TC-spanner (specifically, Corollary 2.3) is nearly tight. The main result of this section, Theorem 4.1, is proved by a construction that adapts ideas of Hesse [19]. At the end of the section, we present a simple direct construction that gives a stronger result for $k = 3$ and $k = 4$: in Lemma 4.5, we demonstrate that the transformation from a 3-TC-spanner to a 2-TC-spanner is asymptotically tight. Since our transformation to 2-TC-spanners incurs the same increase in the number of edges whether we start from 3- or 4-TC-spanners, and since $S_4(G) \leq S_3(G)$, Lemma 4.5 also implies the tightness of the transformation from 4-TC-spanners.

▶ **Theorem 4.1.** *For all $k > 2$ and $\epsilon > 0$, there exists an infinite family of graphs G on n nodes such that $S_2(G)/S_{2k}(G) \geq n^{1-1/k-\epsilon}$.*

Proof. The crux of our construction is the same as Hesse's [19]. The main difference is that he was trying to get an extreme example with a large number of layers, while our goal is to keep the number of layers small. Consequently, Hesse did not analyze our variant of his construction.

▶ **Definition 4.2** (Sets $C_{r,d}$ and $F_{r,d}$, [19])**.**

- $C_{r,d}$ is the set of vectors in $\mathbb{Z}^d$ of Euclidean length at most r.
- $F_{r,d}$ is the set of extremal points, "corners", of the convex hull of $C_{r,d}$.

Now we describe our graph, $G = G(d, r, k)$. It has some similarity to the Butterfly Network: in particular, its nodes are of the form $(\ell, x_1, \ldots, x_k)$ and its edges are of the form $((\ell - 1, x_1, \ldots, x_k), (\ell, x'_1, \ldots, x'_k))$ where ℓ is the node level, an integer in the range from 0 to $2k$. The remaining k coordinates are called *data coordinates*. Like in the Butterfly network, only one data coordinate changes when an edge is traversed, as indicated by the ℓ-coordinate: namely, $x_i = x'_i$ if $i \neq \ell$ and $i \neq k + \ell$. However, the range of data coordinates and the allowed changes are different. In particular, before the first allowed change, each data coordinate has range $C_{r,d}$, after the first change the allowed range is $C_{2r,d}$ and after the second change, the range is $C_{3r,d}$. The allowed change is $x'_i = x_i + z_i$ where $z_i \in F_{r,d}$.

Observe that G is its own $2k$-TC-spanner since every directed path has at most $2k$ edges. That is, $S_{2k}(G) = |G|$. One can also see that the out-degree of every node, except for nodes in layer $2k$, is $|F_{r,d}|$.

In the next lemma, we analyze paths between the first and the last layer of G.

▶ **Lemma 4.3.** *For all $x = (x_1, \ldots, x_k) \in (C_{r,d})^k$ and all $z = (z_1, \ldots, z_k) \in (F_{r,d})^k$, there exists exactly one path from $(0, x)$ to $(2k, x + 2z)$.*

Proof. To analyze possible paths from $(0, x)$ to $(2k, x + 2z)$, we will find all possible values of each data coordinate in the node descriptions on that path. On a path from $(0, x)$ to $(2k, x + 2z)$, we change the i-th data coordinate twice: when we go from layer $i - 1$ to i and when we go from layer $k + i - 1$ to layer $k + i$. Thus, the only possible values of that coordinate are the initial value x_i, the final value $x_i + 2z_i$ and the intermediate value, say, $x_i + u$. Let $v = 2z_i - u$. Then $u, v \in F_{r,d}$ and $(u + v)/2 = z_i$. Because z_i is a corner (an extremal point) of the convex hull of $C_{r,d}$, this implies $u = v = z_i$. Therefore, there is only one possible value of each data coordinate for each node on such a path. ◀

We will call unique paths from $(0, x)$ to $(2k, x + 2z)$, described in the lemma, *straight paths*.

Consider a 2-TC-spanner H. We say that an edge $(u, v) \in H$ is *long* if the difference between the level of v and the level of u is at least k. For every straight path P, say, starting

at some $(0, x)$ and ending at some $(2k, x + 2z)$, there exists some (i, y) such that H contains edges $((0, x), (i, y))$ and $((i, y), (2k, x + 2z))$. We say that the path P is *covered* by these two edges. One of the two edges must be long, and thus it uniquely determines x and z. That is, a long edge can be used to cover at most one straight path. Therefore, H has to contain at least as many long edges as there are straight paths.

Now we give bounds on the numbers $n, |G|$ and s, where n is the number of nodes and s is the number of straight paths in G. Recall that $S_{2k}(G) = |G|$ and that s is a lower bound on $S_2(G)$. We use the following tight bound on the size of $F_{r,d}$ by Bárány and Larman [4].

▶ **Theorem 4.4.** *For every $d \geq 2$ there exist constants $c_1(d), c_2(d)$ such that*

$$c_1(d) r^{d\frac{d-1}{d+1}} \leq |F_{r,d}| \leq c_2(d) r^{d\frac{d-1}{d+1}}.$$

Let $c = |C_{r,d}|$, $a = |C_{3r,d}|/|C_{r,d}|$ and $f = |F_{r,d}|$.

Note that each layer of G has more nodes than the previous. In particular, for sufficiently large r, the size of layer $2k$ is at least $(3/2)^d$ times larger then the size of layer $2k-1$ because the range of the last data coordinate is expanded from $C_{2r,d}$ to $C_{3r,d}$, while the range of the remaining data coordinates remains the same. Therefore, n equals the size of layer $2k$ times a small constant factor, *i.e.,* $n \approx (ac)^k$. Since the out-degree of each node in G is f or 0, $|G| < nf < nac \approx n^{1+1/k}$.

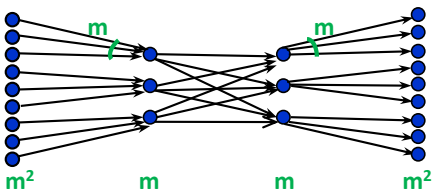
The number of straight paths is $s = (cf)^k$ because a straight path can be specified using any node in layer 0 (c^k choices) and any sequence of the first k edges of the path (f choices for each edge). We will show how to choose r and d, so that $s \geq n^{2-\epsilon}$ or, equivalently, $(cf)^k \geq (ac)^{k(2-\epsilon)}$. It is equivalent to $cf \geq (ac)^{2-\epsilon}$, and finally to $f \geq a^{2-\epsilon} c^{1-\epsilon}$. Since a does not increase with r while c does, we can guarantee $a^2 < c^{\epsilon/2}$ by choosing sufficiently large r. By choosing $d > 4/\epsilon$, we get $f > c^{1-\epsilon/2}$ (by Theorem 4.4). Thus, for sufficiently large r and d, $s \geq n^{2-\epsilon}$.

To summarize, $S_2(G)/S_{2k}(G) \geq s/|G| \geq n^{2-\epsilon}/n^{1+1/k} = n^{1-1/k-\epsilon}$, as stated. ◀

For stretch $k = 3$ (and hence for stretch $k = 4$), a simple construction yields a stronger (tight) lower bound.

▶ **Lemma 4.5.** *There exists an infinite family of graphs G on n nodes with $S_2(G)/S_3(G) = \Omega(\sqrt{n})$.*

Proof. For each m, we construct a 4-layered graph G with m^2 nodes in layers 1 and 4 and m nodes in layers 2 and 3, where the edges are directed from smaller to larger layers and are formed as follows. There is a complete bipartite graph between layers 2 and 3. Each node in layer 2 is connected to m nodes in layer 1, and each node in layer 1 has outdegree 1. The edges between layers 3 and 4 are constructed in the same manner. The resulting graph has $3m^2$ edges and is a 3-TC-spanner. A 2-TC-spanner of this graph must connect m^4 pairs of vertices in layers 1 and 4 via paths of length at most 2. Each shortcut edge can be used by at most m such pairs. Therefore, at least m^3 shortcuts are required. Setting $n = 2m^2 + 2m$, we obtain a graph with a 3-TC-spanner of size $O(n)$, for which every 2-TC-spanner requires $\Omega(n^{3/2})$ edges. ◀



5 Approximation Algorithm for k-TC-Spanner for large k

In this section, we present an algorithm for k-TC-Spanner that provides a better ratio than the algorithm from Theorem 2.1 for stretch $k = \Omega(\log n / \operatorname{loglog} n)$.

▶ **Theorem 5.1.** *k-TC-Spanner can be approximated with ratio $O(n/k^2)$ in polynomial time.*

$O(n/k^2)$-Approximation Algorithm for k-TC-Spanner
Input: directed graph $G = (V, E)$, desired stretch k
1. Let H be a transitive reduction of G.
2. While H contains vertices u, v such that $dist_H(u, v) > k$
   a. Let $(v_1 = u, v_2, ..., v_t = v)$ be the shortest path from u to v in H.
   b. Add a shortcut edge $(v_{\lfloor k/4 \rfloor}, v_{t-\lfloor k/4 \rfloor})$ to H.
3. Output H.

Proof. To prove that the above algorithm has the desired approximation ratio, it is enough to analyze it on each weakly connected component. Let OPT be the size of the sparsest k-TC-spanner on a weakly connected component with n nodes. Then $OPT \geq TR(G) \geq n-1$ where $TR(G)$ denotes the size of the transitive reduction of G. We will show that $O(n^2/k^2)$ shortcut edges are added to H, giving the desired ratio.

Observe that each shortcut edge decreases the distance from at least $k/2$ to less than $k/2$ for $\Omega(k^2)$ pairs of vertices (v_i, v_j) on the shortest path from u to v: namely, all pairs for which $i \leq \lfloor k/4 \rfloor$ and $j \geq t - \lfloor k/4 \rfloor$. Since initially there are $O(n^2)$ pairs of vertices at distance larger than $k/2$, only $O(n^2/k^2)$ shortcut with this property can be added. This completes the proof of the lemma. ◀

References

1 Alfred V. Aho, M. R. Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM J. Comput.*, 1(2):131–137, 1972.
2 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.
3 Baruch Awerbuch. Communication-time trade-offs in network synchronization. In *PODC*, pages 272–276, 1985.
4 Imre Bárány and David Larman. The convex hull of the integer points in a large ball. *Mathematische Annalen*, 312:167–181, 1998.
5 Surender Baswana and Sandeep Sen. Approximate distance oracles for unweighted graphs in expected $\tilde{O}(n^2)$ time. *ACM Transactions on Algorithms*, 2(4):557–577, 2006.
6 Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyomin Jung, Sofya Raskhodnikova, and David Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. In *RANDOM*, pages 448–461, 2010.
7 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David Woodruff. Transitive-closure spanners. In *SODA*, pages 932–941, 2009.
8 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David Woodruff. Transitive-closure spanners of the hypercube and the hypergrid. ECCC Report TR09-046, 2009.
9 Arnab Bhattacharyya, Elena Grigorescu, Sofya Raskhodnikova, and David Woodruff. Steiner transitive-closure spanners of d-dimensional posets. Manuscript, 2010.
10 Hanls L. Bodlaender, Gerard Tel, and Nicola Santoro. Tradeoffs in non-reversing diameter. *Nordic Journal of Computing*, 1(1):111 – 134, 1994.

11 Edith Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. *SIAM J. Comput.*, 28(1):210–236, 1998.

12 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *JACM*, 47(1):132–166, 2000.

13 Lenore Cowen. Compact routing with minimum stretch. *J. Algorithms*, 38(1):170–183, 2001.

14 Lenore Cowen and Christopher G. Wagner. Compact roundtrip routing in directed networks. *J. Algorithms*, 50(1):79–95, 2004.

15 M. Elkin. Computing almost shortest paths. In *PODC*, pages 53–62, 2001.

16 Michael Elkin and David Peleg. Strong inapproximability of the basic k-spanner problem. In *ICALP*, pages 636–647, 2000.

17 Michael Elkin and David Peleg. The client-server 2-spanner problem with applications to network design. In *SIROCCO*, pages 117–132, 2001.

18 Michael Elkin and David Peleg. The hardness of approximating spanner problems. *Theory Comput. Syst.*, 41(4):691–729, 2007.

19 William Hesse. Directed graphs requiring large numbers of shortcuts. In *SODA*, pages 665–669, 2003.

20 D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, Boston, 1997.

21 G. Kortsarz and D. Peleg. Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994.

22 Guy Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

23 David Peleg. *Distributed computing: a locality-sensitive approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

24 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

25 David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.

26 David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. *JACM*, 36(3):510–530, 1989.

27 Sofya Raskhodnikova. Transitive-closure spanners: a survey. In Oded Goldreich, editor, *Property Testing*, volume LNCS 6390, pages 167–196. Springer, Heidelberg, 2010.

28 Mikkel Thorup. On shortcutting digraphs. In *Graph-Theoretic Concepts in Computer Science*, volume 657, pages 205–211, 1993.

29 Mikkel Thorup. Shortcutting planar digraphs. *Combinatorics, Probability & Computing*, 4:287–315, 1995.

30 Mikkel Thorup. Parallel shortcutting of rooted trees. *J. Algorithms*, 23(1):139–159, 1997.

31 Mikkel Thorup and Uri Zwick. Compact routing schemes. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 1–10, 2001.

32 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *JACM*, 52(1):1–24, 2005.

33 Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, New York, NY, USA, 2007.