Lower Bounds for Local Monotonicity Reconstruction from Transitive-Closure Spanners*

Arnab Bhattacharyya† Elena Grigorescu† Madhav Jha‡ Kyomin Jung§
Sofya Raskhodnikova‡ 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. Transitive-closure spanners are a common abstraction for applications in access control, property testing and data structures.

We show a connection between 2-TC-spanners and *local monotonicity reconstructors*. A local monotonicity reconstructor, introduced by Saks and Seshadhri (SIAM Journal on Computing, 2010), is a randomized algorithm that, given access to an oracle for an almost monotone function $f : [m]^d \to \mathbb{R}$, can quickly evaluate a related function $g : [m]^d \to \mathbb{R}$ which is guaranteed to be monotone. Furthermore, the reconstructor can be implemented in a distributed manner. We show that an efficient local monotonicity reconstructor implies a sparse 2-TC-spanner of the directed hypergrid (hypercube), providing a new technique for proving lower bounds for local monotonicity reconstructors. Our connection is, in fact, more general: an efficient local monotonicity reconstructor for functions on any partially ordered set (poset) implies a sparse 2-TC-spanner of the directed acyclic graph corresponding to the poset.

We present tight upper and lower bounds on the size of the sparsest 2-TC-spanners of the directed hypercube and hypergrid. These bounds imply tighter lower bounds for local monotonicity reconstructors that nearly match the known upper bounds.

1 Introduction

Graph spanners were introduced in the context of distributed computing [16], and since then have found numerous applications, such as efficient routing [10, 11, 18, 19, 25], simulating synchronized protocols in unsynchronized networks [17], parallel and distributed algorithms for approximating shortest paths [8, 9, 13], and algorithms for distance oracles [4, 26]. Several variants on graph spanners have been defined. In this work, we focus on *transitive-closure* spanners that were introduced in [6] as a common abstraction for applications in access control, property testing and data structures.

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)** of G 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.*

*The preliminary version of the paper appeared in RANDOM 2010 [5].

†Massachusetts Institute of Technology, USA. Email:{abhatt,elena_g}@mit.edu. AB supported in part by a DOE Computational Science Graduate Fellowship and NSF Awards 0514771, 0728645, and 0732334.

‡Pennsylvania State University, USA. Email: {mxj201,sofya}@cse.psu.edu. Supported by NSF CAREER award 0845701.

§KAIST, South Korea. Email:kyomin@kaist.edu

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

2. *For all vertices $u, v \in V$, if $d_G(u,v) < \infty$, then $d_H(u,v) \leq k$.*

Thus, a k-transitive-closure-spanner (or k-TC-spanner) is a graph with small diameter[1] that preserves the connectivity of the original graph. In the applications above, the goal is to find the sparsest k-TC-spanner for a given k and G. The number of edges in the sparsest k-TC-spanner of G is denoted by $S_k(G)$.

Our Contributions The contributions of this work fall into two categories: (1) We show that an efficient local monotonicity reconstructor implies a sparse 2-TC-spanner of the directed hypergrid (hypercube), providing a new technique for proving lower bounds for local monotonicity reconstructors. (2) We present tight upper and lower bounds on the size of the sparsest 2-TC-spanners of the directed hypercube and hypergrid. These bounds imply tighter lower bounds for local monotonicity reconstructors for these graphs that nearly match the upper bounds given in [20].

1.1 Our Lower Bounds for Local Monotonicity Reconstruction

Property-preserving data reconstruction was introduced in [1]. In this model, a reconstruction algorithm, called a *filter*, sits between a *client* and a *dataset*. A dataset is viewed as a function $f : \mathcal{D} \to \mathcal{R}$. The client accesses the dataset using *queries* of the form $x \in \mathcal{D}$ to the filter. The filter *looks up* a small number of values in the dataset and outputs $g(x)$, where g must satisfy some fixed *structural* property $\mathcal{P}$. Extending this notion, Saks and Seshadhri [20] defined *local* reconstruction. A filter is *local* if it allows for a local (or distributed) implementation: namely, if the output function g does not depend on the order of the queries.

Definition 1.2 (Local filter)**.** *A* local filter *for reconstructing property $\mathcal{P}$ is an algorithm A that has oracle access to a function $f : \mathcal{D} \to \mathcal{R}$, and to an auxiliary random string ρ (the "random seed"), and takes as input $x \in \mathcal{D}$. For fixed f and ρ, A runs deterministically on input x to produce an output $A_{f,\rho}(x) \in \mathcal{R}$. (Note that a local filter has no internal state to store previously made queries.) The function $g(x) = A_{f,\rho}(x)$ output by the filter must satisfy the following conditions:*

- *For each f and ρ, the function g must satisfy $\mathcal{P}$.*
- *If f satisfies $\mathcal{P}$, then g must be identical to f with probability at least $1-\delta$, for some* error probability *$\delta \leq 1/3$. The probability is taken over ρ.*

In answering query $x \in \mathcal{D}$, the filter A may ask for values of f at domain points of its choice (possibly adaptively) using its oracle access to f. Each such access made to the oracle is called a *lookup* to distinguish it from the client query x. A local filter is *non-adaptive* if the set of domain points that the filter looks up to answer an input query x does not depend on answers given by the oracle.

In [20], the authors also required that g must be sufficiently close to f: *With high probability (over the choice of ρ), $Dist(g,f) \leq B(n) \cdot Dist(f,\mathcal{P})$, where $B(n)$ is called the* error blow-up. ($Dist(g,f)$ is the number of points in the domain on which f and g differ. $Dist(f,\mathcal{P})$ is $\min_{g \in \mathcal{P}} Dist(g,f)$.) If a local filter along with Definition 1.2 satisfies this condition, we call it *distance-respecting*.

1.1.1 Local Monotonicity Reconstructors

The most studied property in the local reconstruction model is monotonicity of functions [20, 1].

To define monotonicity of functions, consider an n-element poset V_n 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 $f(x) \leq f(y)$

[1] The definition of diameter used in this work and other papers on transitive-closure spanners 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.

for all $(x, y) \in E$. We focus on posets whose relation graph is a *directed hypergrid*. The *directed hypergrid*, denoted $\mathcal{H}_{m,d}$, has vertex set $\{1, 2, \ldots, m\}^d$ and edge set $\{(x, y) : \exists$ unique $i \in \{1, \ldots, d\}$ such that $y_i - x_i = 1$ and for $j \neq i, y_j = x_j\}$. For the special case $m = 2$, $\mathcal{H}_{2,d}$ is called the *hypercube* and is also denoted by $\mathcal{H}_d$. A monotonicity filter needs to ensure that the output function g is monotone. For instance, if G_n is a directed line, $\mathcal{H}_{n,1}$, the filter needs to ensure that the output sequence specified by g is sorted.

To motivate monotonicity reconstructors for hypergrids, consider the scenario of rolling admissions: An admissions office assigns d scores to each application, such as the applicant's GPA, SAT results, essay quality, etc. Based on these scores, some complicated (third-party) algorithm outputs the probability that a given applicant should be accepted. The admissions office wants to make sure "on the fly" that strictly better applicants are given higher probability, that is, probabilities are *monotone* in scores. A hypergrid monotonicity filter may be used here.

A local filter can be implemented in a distributed manner with an additional guarantee that every copy of the filter will correct to the same monotone function of the scores. This can be done by supplying the same random seed to each copy of the filter.

[20] gives a *distance-respecting* local monotonicity filter for the directed hypergrid, $\mathcal{H}_{m,d}$, that makes $(\log m)^{O(d)}$ lookups per query. No non-trivial monotonicity filter for the hypercube $\mathcal{H}_d$ (performing $o(2^d)$ lookups per query) is known. One of the monotonicity filters in [1] is a local filter for the directed line $\mathcal{H}_{m,1}$ with $O(\log m)$ lookups per query (but a worse error blow up than in [20]). As observed in [20], this upper bound is tight. [20] also presents a lower bound of $2^{\alpha d}$ on the number of lookups per query for a *distance-respecting* local monotonicity filter on $\mathcal{H}_d$ with *error blow-up* $2^{\beta d}$, where α, β are sufficiently small constants. Notably, all known local monotonicity filters are *non-adaptive*.

We show how to construct sparse 2-TC-spanners from local monotonicity reconstructors with low lookup complexity. These constructions, together with our lower bounds on the size of 2-TC-spanners of the hypergrid and hypercube (Section 1.2), imply lower bounds on lookup complexity of local monotonicity reconstructors for these graphs with arbitrary error blow-up. We state our transformations from non-adaptive and adaptive reconstructors separately.

Theorem 1.1 (Transformation from non-adaptive Local Monotonicity Reconstructors to 2-TC-spanners)**.** *Let $G_n = (V_n, E)$ be a poset on n nodes. Suppose there is a* non-adaptive *local monotonicity reconstructor A for G_n that looks up at most $\ell(n)$ values on any query and has* error *probability at most δ. Then there is a 2-TC-Spanner of G_n with $O(n\ell(n) \cdot \lceil \log n / \log(1/\delta) \rceil)$ edges.*

Next theorem applies even to *adaptive* local monotonicity reconstuctors. It takes into account how many lookups on query x are nodes *incomparable* to x. (Two nodes x, y are called *comparable* if x is reachable from y or y is reachable from x; otherwise, they are *incomparable*.) In particular, if there are no such lookups then the constructed 2-TC-spanner is of the same size as in Theorem 1.1.

Theorem 1.2 (Transformation from adaptive Local Monotonicity Reconstructors to 2-TC-spanners)**.** *Let $G_n = (V_n, E)$ be a poset on n nodes. Suppose there is an (*adaptive*) local monotonicity reconstructor A for G_n that, for any query $x \in V_n$, looks up at most $\ell_1(n)$ vertices comparable to x and at most $\ell_2(n)$ vertices incomparable to x, and has* error *probability at most δ. Then there is a 2-TC-Spanner of G_n with $O(n\ell_1(n) \cdot 2^{\ell_2(n)} \lceil \log n / \log(1/\delta) \rceil)$ edges.*

In Theorem 1.1 and 1.2, when δ is sufficiently small, the bounds on the 2-TC-Spanner size become $O(n\ell(n))$ and $O(n\ell_1(n) \cdot 2^{\ell_2(n)})$, respectively.

As mentioned earlier, all known monotonicity reconstructors are non-adaptive. It is an open question whether it is possible to give a transformation from adaptive local monotonicity reconstructors to 2-TC-spanners without incurring an exponential dependence on the number of lookups made to points incomparable to the query point. We do not know whether this dependence is an artifact of the proof or an indication that lookups to incomparable points might be helpful for adaptive local monotonicity reconstructors.

In Theorems 1.5 and 1.6 (Section 1.2), we present nearly tight bounds on the size of the sparsest 2-TC-spanners of the hypercube and the hypergrid. Theorems 1.1 and 1.2, together with the lower bounds in Theorems 1.5 and 1.6, imply the following lower bounds on the lookup complexity of local monotonicity reconstructors for these graphs with arbitrary error blow-up.

Corollary 1.3. *Consider a nonadaptive local monotonicity filter with constant error probability δ. If the filter is for functions $f : \mathcal{H}_{m,d} \to \mathbb{R}$, it must perform $\Omega\left(\frac{\log^{d-1} m}{d^d(2\log\log m)^{d-1}}\right)$ lookups per query. If the filter is for functions $f : \mathcal{H}_d \to \mathbb{R}$, it must perform $\Omega\left(2^{\alpha d}/d\right)$ lookups per query, where $\alpha \geq 0.1620$.*

Corollary 1.4. *Consider an (adaptive) local monotonicity filter with constant error probability δ, that for every query $x \in V_n$, looks up at most ℓ_2 vertices incomparable to x. If the filter is for functions $f : \mathcal{H}_{m,d} \to \mathbb{R}$, it must perform $\Omega\left(\frac{\log^{d-1} m}{2^{\ell_2} d^d(2\log\log m)^{d-1}}\right)$ lookups to vertices comparable to x per query x. If the filter is for functions $f : \mathcal{H}_d \to \mathbb{R}$, it must perform $\Omega\left(2^{\alpha d-\ell_2}/d\right)$ comparable lookups, where $\alpha \geq 0.1620$.*

Prior to this work, no lower bounds for monotonicity reconstructors on $\mathcal{H}_{m,d}$ with dependence on both m and d were known. Unlike the bound in [20], our lower bounds hold for any error blow-up and for non-distance-respecting filters. Our bounds are tight for non-adaptive reconstructors. Specifically, for the hypergrid $\mathcal{H}_{m,d}$ of constant dimension d, the number of lookups is $(\log m)^{\Theta(d)}$, and for the hypercube $\mathcal{H}_d$, it is $2^{\Theta(d)}$ for any error blow-up.

1.1.2 Testers vs. Reconstructors

[6] obtained monotonicity testers from 2-TC-spanners. Unlike in the application to monotonicity testing, here we use *lower bounds* on the size of 2-TC-spanners to prove *lower bounds* on complexity of local monotonicity reconstuctors. Lower bounds on the size of 2-TC-spanners do not imply corresponding lower bounds on monotonicity testers. *E.g.,* the best monotonicity tester on $\mathcal{H}_d$ runs in $O(d^2)$ time [14, 12], while, as shown in Theorem 1.6, every 2-TC-spanner of $\mathcal{H}_d$ must have size exponential in d.

1.2 Our Results on 2-TC-Spanners of the Hypercube and Hypergrid

Our main theorem, proved in Section 4, gives a set of explicit bounds on $S_2(\mathcal{H}_{m,d})$:

Theorem 1.5 (Hypergrid)**.** *Let $S_2(\mathcal{H}_{m,d})$ denote the number of edges in the sparsest 2-TC-spanner of $\mathcal{H}_{m,d}$. Then[2] for $m \geq 3$,*

$$\Omega\left(\frac{m^d \log^d m}{(2d\log\log m)^{d-1}}\right) \;=\; S_2(\mathcal{H}_{m,d}) \;\leq\; m^d \log^d m.$$

The upper bound in Theorem 1.5 follows from a general construction of k-TC-spanners for graph products for arbitrary $k \geq 2$, presented in Section 4.1. The lower bound is the most technically difficult part of our work. It is proved by a reduction of the 2-TC-spanner construction for $[m]^d$ to that for the $2 \times [m]^{d-1}$ grid and then directly analyzing the number of edges required for a 2-TC-spanner of $2 \times [m]^{d-1}$. We show a tradeoff between the number of edges in the 2-TC-spanner of the $2 \times [m]^{d-1}$ grid that stay within the hyperplanes $\{1\} \times [m]^{d-1}$ and $\{2\} \times [m]^{d-1}$ versus the number of edges that cross from one hyperplane to the other. The proof proceeds in multiple stages. Assuming an upper bound on the number of edges staying within the hyperplanes, each stage is shown to contribute a substantial number of new edges crossing between the hyperplanes. The proof of this tradeoff lemma is already non-trivial for $d = 2$ and is presented first in Section 4.2.1. The proof for $d > 2$ is presented in Section 4.2.2.

[2]Logarithms are always base 2 unless otherwise indicated.

While Theorem 1.5 is most useful when m is large and d is small, in Theorem 6.1, we present bounds on $S_2(\mathcal{H}_{m,d})$ which are optimal up to a factor of d^{2m} and, thus, supersede the bounds from Theorem 1.5 when m is small. In particular, for constant m, our upper and lower bounds differ only by a $\mathrm{poly}(d)$ factor. The general form of these bounds is a somewhat complicated combinatorial expression but they can be estimated numerically. Specifically, $S_2(\mathcal{H}_{m,d}) = 2^{c_m d}\,\mathrm{poly}(d)$, where $c_2 \approx 1.1620$, $c_3 \approx 2.03$, $c_4 \approx 2.82$ and $c_5 \approx 3.24$, each significantly smaller than the exponents corresponding to the transitive closure sizes for the appropriate m.

We first consider the special case of $m = 2$ (i.e., the hypercube) in Section 5 and then generalize the arguments to larger m in Section 6. Specifically, we obtain the following theorem for the hypercube.

Theorem 1.6 (Hypercube)**.** *Let $S_2(\mathcal{H}_d)$ be the number of edges in the sparsest 2-TC-spanner of $\mathcal{H}_d$. Then $\Omega(2^{cd}) = S_2(\mathcal{H}_d) = O(d^3 2^{cd})$, where $c \approx 1.1620$.*

We prove the theorem by giving nearly matching upper and lower bounds on $S_2(\mathcal{H}_d)$ in terms of an expression with binomial coefficients, and later numerically estimating the value of the expression. We prove the upper bound in Theorem 1.6 by presenting a randomized construction of a 2-TC-spanner of the directed hypercube. Curiously, even though the upper and lower bounds above differ by a factor of $O(d^3)$, we can show that our construction yields a 2-TC-spanner of $\mathcal{H}_d$ of size within $O(d^2)$ of the optimal.

As a comparison point for our bounds, note that the obvious bounds on $S_2(\mathcal{H}_d)$ are the number of edges in the d-dimensional hypercube, $2^{d-1}d$, and the number of edges in the transitive closure of $\mathcal{H}_d$, which is $3^d - 2^d$. (An edge in the transitive closure of $\mathcal{H}_d$ has 3 possibilities for each coordinate: both endpoints are 0, both endpoints are 1, or the first endpoint is 0 and the second is 1. This includes self-loops, so we subtract the number of vertices in $\mathcal{H}_d$ to get the desired quantity.) Thus, $2^{d-1}d \le S_2(\mathcal{H}_d) \le 3^d - 2^d$. Similarly, the straightforward bounds on the number of edges in a 2-TC-spanner of $\mathcal{H}_{m,d}$ in terms of the number of edges in the directed grid and in its transitive closure are $dm^{d-1}(m-1)$ and $\left(\frac{m^2+m}{2}\right)^d - m^d$, respectively.

1.2.1 Previous Work on Bounding S_k for other Families of Graphs

Thorup [22] considered a special case of TC-spanners of graphs G that have at most twice as many edges as G, and conjectured that for all directed graphs G on n nodes there are such k-TC-spanners with k polylogarithmic in n. He proved this for planar graphs [23], but later Hesse [15] gave a counterexample to Thorup's conjecture for general graphs. For all small $\epsilon > 0$, he constructed a family of graphs with $n^{1+\epsilon}$ edges for which all n^{ϵ}-TC-spanners require $\Omega(n^{2-\epsilon})$ edges. TC-spanners were studied for directed trees: implicitly in [12, 2, 3, 7, 27] and explicitly in [24]. For the directed line, [2] (and later, [3]) expressed $S_k(\mathcal{H}_{n,1})$ in terms of the inverse Ackermann function. (See Section 2.2 for a definition.)

Lemma 1.7 ([2, 3, 6])**.** *Let $S_k(\mathcal{H}_{n,1})$ denote the number of edges in the sparsest k-TC-spanner of the directed line $\mathcal{H}_{n,1}$. Then $S_2(\mathcal{H}_{n,1}) = \Theta(n \log n)$, $S_3(\mathcal{H}_{n,1}) = \Theta(n \log\log n)$, $S_4(\mathcal{H}_{n,1}) = \Theta(n \log^* n)$ and, more generally, $S_k(\mathcal{H}_{n,1}) = \Theta(n\lambda_k(n))$ where $\lambda_k(n)$ is the inverse Ackermann function.*

The same bound holds for directed trees [2, 7, 24]. An $O(n \log n \cdot \lambda_k(n))$ bound on S_k for H-minor-free graph families (e.g., bounded genus and bounded tree-width graphs) was given in [6].

2 Preliminaries

2.1 Notation

For a positive integer m, we denote $\{1, \ldots, m\}$ by $[m]$. For $x \in \{0,1\}^d$, we use $|x|$ to denote the weight of x, that is, the number of non-zero coordinates in x. Level i in a hypercube contains all vertices of weight

i. The partial order $\preceq$ on the hypergrid $\mathcal{H}_{m,d}$ is defined as follows: $x \preceq y$ for two vertices $x, y \in [m]^d$ iff $x_i \leq y_i$ for all $i \in [d]$. Similarly, $x \prec y$, if x and y are distinct vertices in $[m]^d$ satisfying $x \preceq y$. Vertices x and y are *comparable* if either y is *above* x (that is, $x \preceq y$) or y is *below* x (that is, $y \preceq x$). We denote a path from v_1 to v_ℓ, consisting of edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{\ell-1}, v_\ell)$ by $(v_1, \ldots, v_\ell)$.

2.2 The Inverse Ackermann Hierarchy

Our definition of inverse Ackermann functions is derived from the discussion in [21]. For a given function $f : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ such that $f(x) < x$ for all $x > 2$, define the function $f^*(x) : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ to be the following:

$$f^*(x) = \min\{k \in \mathbb{Z}^{\geq 0} : f^{(k)}(x) \leq 2\}, \text{ where } f^{(k)} \text{ denotes } f \text{ composed with itself } k \text{ times}$$

We note that the solution to the following recursion:

$$T(n) \leq \begin{cases} 0 & \text{if } n \leq 2 \\ a \cdot n + \frac{n}{f(n)} \cdot T(f(n)) & \text{if } n > 2 \end{cases}$$

is $T(n) = a \cdot n \cdot f^*(n)$. This follows from the fact that $f^*(f(n)) = f^*(n) - 1$ for $n > 2$.

We define the inverse Ackermann hierarchy to be a sequence of functions $\lambda_k(\cdot)$ for $k \geq 0$. As the base cases, we have $\lambda_0(n) = n/2$ and $\lambda_1(n) = \sqrt{n}$. For $j \geq 2$, we define $\lambda_j(n) = \lambda^*_{j-2}(n)$. Thus, $\lambda_2(n) = \Theta(\log n)$, $\lambda_3(n) = \Theta(\log \log n)$ and $\lambda_4(n) = \Theta(\log^* n)$. Note that the $\lambda_k(\cdot)$ functions defined here coincide (upto constant additive differences) with the $\lambda(k, \cdot)$ functions in [2] although they were formulated a bit differently there.

Finally, we define the inverse Ackermann function $\alpha(\cdot)$ to be $\alpha(n) = \min\{k \in \mathbb{Z}^{\geq 0} : \lambda_{2k}(n) \leq 3\}$.

3 Transformation from Local Monotonicity Reconstructors to 2-TC-Spanners

In this section, we prove Theorems 1.1 and 1.2.

3.1 From Non-Adaptive Local Monotonicity Reconstructors to 2-TC-Spanners

Proof of Theorem 1.1. Let A be a local reconstructor given by the statement of the theorem. Let $\mathcal{F}$ be the set of pairs (x, y) with x, y in V_n such that $x \prec y$. Then, $\mathcal{F}$ is of size at most $\binom{n}{2}$. Given $(x, y) \in \mathcal{F}$, let $\mathsf{cube}(x, y)$ be the set $\{z \in V_n : x \preceq z \preceq y\}$. Define function $f^{(x,y)}(v)$ to be 1 on all $v \succeq x$ and all $v \succeq y$, and 0 everywhere else. Also, define function $f^{(\overline{x,y})}(v)$, which is identical to $f^{(x,y)}(v)$ for all $v \notin \mathsf{cube}(x, y)$ and 0 for $v \in \mathsf{cube}(x, y)$. Both, $f^{(x,y)}$ and $f^{(\overline{x,y})}$, are monotone functions for all $(x, y) \in \mathcal{F}$. Let A_ρ be the deterministic algorithm which runs A with the random seed fixed to ρ. We say a string ρ is *good* for $(x, y) \in \mathcal{F}$ if filter A_ρ on input $f^{(x,y)}$ returns $g = f^{(x,y)}$ *and* on input $f^{(\overline{x,y})}$ returns $g = f^{(\overline{x,y})}$.

Now we show that there exists a set S of size $s \leq \lceil 2 \log n / \log(1/2\delta) \rceil$, consisting of strings used as random seeds by A, such that for every $(x, y) \in \mathcal{F}$ some string $\rho \in S$ is good for (x, y). We choose S by picking strings used as random seeds uniformly and independently at random. Since A has error probability at most δ, we know that for every monotone f, with probability at least $1 - \delta$ (with respect to the choice of ρ), the function $A_{f,\rho}$ is identical to f. Then, for fixed $(x, y) \in \mathcal{F}$ and uniformly random ρ,

$$\begin{aligned} \Pr[\rho \text{ is not } \textit{good} \text{ for } (x, y)] \leq \quad & \Pr[A_\rho \text{ on input } f^{(x,y)} \text{ fails to output } f^{(x,y)}] \\ + \quad & \Pr[A_\rho \text{ on input } f^{(\overline{x,y})} \text{ fails to output } f^{(\overline{x,y})}] \leq 2\delta. \end{aligned}$$

Since strings in S are chosen independently, $\Pr[\text{no } \rho \in S \text{ is good for } (x,y)] \le (2 \cdot \delta)^s$, which, for $s = \lceil 2 \log n / \log(1/2\delta) \rceil$, is at most $1/n^2 < 1/|\mathcal{F}|$. By a union bound over $\mathcal{F}$,

$$\Pr[\text{for some } (x,y) \in \mathcal{F}, \text{ no } \rho \in S \text{ is good for } (x,y)] < 1.$$

Thus, there exists a set S with required properties.

We construct our 2-TC-spanner $H = (V_n, E_H)$ of G_n using set S described above. Let $\mathcal{N}_\rho(x)$ be the set consisting of x and all vertices looked up by A_ρ on query x. (Note that the set $\mathcal{N}_\rho(x)$ is well-defined since algorithm A is assumed to be *non-adaptive*). For each string $\rho \in S$ and each vertex $x \in V_n$, connect x to all comparable vertices in $\mathcal{N}_\rho(x)$ (other than itself) and orient these edges according to their direction in G_n.

We prove H is a 2-TC-Spanner as follows. Suppose not, *i.e.,* there exists $(x,y) \in \mathcal{F}$ with no path of length at most 2 in H from x to y. Consider $\rho \in S$ which is *good* for (x,y). Define function h by setting $h(v) = f^{(x,y)}(v)$ for all $v \notin \mathsf{cube}(x,y)$. Then $h(v) = f^{(\overline{x,y})}(v)$ for all $v \notin \mathsf{cube}(x,y)$, by definition of $f^{(\overline{x,y})}$. For a $v \in \mathsf{cube}(x,y)$, set $h(v)$ to 1 for $v \in \mathcal{N}_\rho(x)$ and to 0 for $v \in \mathcal{N}_\rho(y)$. All unassigned points are set to 0. By the assumption above, $\mathcal{N}_\rho(x) \cap \mathcal{N}_\rho(y)$ does not contain any points in $\mathsf{cube}(x,y)$. Therefore, h is well-defined. Since ρ is *good* for (x,y) and h is identical to $f^{(x,y)}$ for all lookups made on query x, $A_\rho(x) = h(x) = 1$. Similarly, $A_\rho(y) = h(y) = 0$. But $x \prec y$, so $A_{h,\rho}(v)$ is not monotone. Contradiction.

The number of edges in H is at most

$$\sum_{x \in V_n, \rho \in S} |\mathcal{N}_\rho(x)| \le n \cdot \ell(n) \cdot s \le n\ell(n) \cdot \lceil 2 \log n / \log(1/2\delta) \rceil. \qquad \square$$

3.2 From Adaptive Local Monotonicity Reconstructors to 2-TC-Spanners

The complication in the transformation from an adaptive filter is that the set of vertices looked up by the filter depends on the oracle that the filter is invoked on.

Proof of Theorem 1.2. Define $\mathcal{F}$, $f^{(x,y)}$, $f^{(\overline{x,y})}$, A_ρ and S as in the proof of Theorem 1.1. As before, for each $x \in V_n$, we define sets $\mathcal{N}_\rho(x)$, and construct the 2-TC-Spanner H by connecting each x to comparable points in $\mathcal{N}_\rho(x)$ for all $\rho \in S$ and orienting the edges according to G_n. However, now $\mathcal{N}_\rho(x)$ is a union of several sets $\mathcal{N}_\rho^{b,w}(x)$, indexed by $b \in \{0,1\}$ and $w \in \{0,1\}^{\ell_2(n)}$. (In addition, $\mathcal{N}_\rho(x)$ contains x.) For each $x \in V_n$, $b \in \{0,1\}$ and $w \in \{0,1\}^{\ell_2(n)}$, let $\mathcal{N}_\rho^{b,w}(x) \subseteq V_n$ be the set of lookups performed by A_ρ on query x, assuming that the oracle answers all lookups as follows. When a lookup y is comparable to x, answer 0 if $y \prec x$, b if $y = x$, 1 if $x \prec y$. Otherwise, if y is the i'th lookup made to an incomparable point for some $i \in [\ell_2]$, answer $w[i]$. Recall that we set $N_\rho(x)$ to be the union of $N_\rho^{b,w}$ for all $b \in \{0,1\}$ and all $w \in \{0,1\}^{\ell_2(n)}$. This completes the description of $\mathcal{N}_\rho(x)$ and construction of H.

The argument that H is a 2-TC-spanner proceeds similarly to that in the proof of Theorem 1.1. The caveat is that an adaptive local filter might choose lookups based on the answers to previous lookups. The constructed function h sets $h(x) = 1$ and $h(y) = 0$. Further all points comparable to x are set to 0 if they are below x and 1 if they are above x. However, points incomparable to x might be comparable to y and are set to 0 or 1, depending on whether they are above or below y. Since we included sets of points queried under all these possibilities in $\mathcal{N}_\rho(x)$, we can now conclude that $A_\rho(x) = h(x) = 1$. The same applies for y. So, $A_{h,\rho}$ outputs a non-monotone function, witnessed by the pair (x,y). Contradiction.

We proceed to bound the number of edges E_H in H. For each $\rho \in S$, $x \in V_n$, $b \in \{0,1\}$, and $w \in \{0,1\}^{\ell_2(n)}$, the number of vertices in $N_{b,w}^{\rho}(x)$ comparable to x is at most $\ell_1(n)$. Therefore,

$$|E_H| \le \ell_1(n) \cdot 2 \cdot 2^{\ell_2(n)} \cdot |S| \le O\left(n \cdot \ell_1(n) \cdot 2^{\ell_2(n)} \lceil \log n / \log(1/\delta) \rceil\right). \qquad \square$$

4 2-TC-Spanners of Low-Dimensional Hypergrids

In this section, we describe the proof of Theorem 1.5 which gives explicit bounds on the size of the sparsest 2-TC-spanner for $\mathcal{H}_{m,d}$. The upper bound in Theorem 1.5 is proved in Section 4.1 and lower bound in Section 4.2.

4.1 Upper Bound

The upper bound in Theorem 1.5 follows straightforwardly from a more general statement about TC-spanners of product graphs presented in the following section. In the same section, we derive the upper bound in Theorem 1.5.

4.1.1 Construction for Product Graphs

This section explains how to construct a TC-spanner of the Cartesian product of graphs G_1 and G_2 from TC-spanners of G_1 and G_2. Since the directed hypergrid is the Cartesian product of directed lines, and optimal TC-spanner constructions are known for the directed line, our construction yields sparse TC-spanners for the grid (Corollary 4.2). We start by defining two graph products: Cartesian and strong.

Definition 4.1 (Graph products)**.** *Given graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a product of G_1 and G_2 is a new graph G with vertex set $V_1 \times V_2$. For the* Cartesian graph product*, denoted by $G_1 \times G_2$, graph G contains an edge from (u_1, u_2) to (v_1, v_2) if and only if $u_1 = v_1$ and $(u_2, v_2) \in E_2$, or $(u_1, v_1) \in E_1$ and $u_2 = v_2$. For the* strong graph product*, denoted by $G_1 \circ G_2$, graph G contains an edge from (u_1, u_2) to (v_1, v_2) if and only if $u_1 = v_1$ and $(u_2, v_2) \in E_2$, or $(u_1, v_1) \in E_1$ and $u_2 = v_2$, or $(u_1, v_1) \in E_1$ and $(u_2, v_2) \in E_2$.*

For example, $\mathcal{H}_{m,2} = \mathcal{H}_{m,1} \times \mathcal{H}_{m,1}$ and $\text{TC}(\mathcal{H}_{m,2}) = \text{TC}(\mathcal{H}_{m,1}) \circ TC(\mathcal{H}_{m,1})$, where $\text{TC}(G)$ denotes the transitive closure of G.

Lemma 4.1. *Let G_1 and G_2 be directed graphs with k-TC-spanners S_1 and S_2, respectively. Then $S_1 \circ S_2$ is a k-TC-spanner of $G = G_1 \times G_2$.*

Proof. Suppose (u, v) and (u', v') are comparable vertices in $G_1 \times G_2$. Then, by definition of the Cartesian product, $u \preceq u'$ in G_1 and $v \preceq v'$ in G_2. Let $(u_1, u_2, \dots, u_\ell)$ be the shortest path in S_1 from $u = u_1$ to $u' = u_\ell$, and $(v_1, v_2, \dots, v_t)$ the shortest path in S_2 from $v = v_1$ to $v' = v_t$. Assume w.t.o.g. that $l \leq t$. Then $((u_1, v_1), (u_2, v_2), \dots, (u_\ell, v_\ell) \dots, (u_\ell, v_t))$ is a path in $S_1 \circ S_2$ of length $t \leq k$, from (u, v) to (u', v'). Therefore, $S_1 \circ S_2$ is a k-TC-spanner of $G = G_1 \times G_2$. □

Lemma 4.1 together with previous results on the size of k-TC-spanners for the line $\mathcal{H}_{m,1}$, summarized in Lemma 1.7, imply an upper bound on the size of a k-TC-spanner of the directed hypergrid $\mathcal{H}_{m,d}$:

Corollary 4.2. *Let $S_k(\mathcal{H}_{m,d})$ denote the number of edges in the sparsest k-TC-spanner of the directed d-dimensional hypergrid $\mathcal{H}_{m,d}$. Then $S_k(\mathcal{H}_{m,d}) = O(m^d \lambda_k(m)^d c^d)$ for appropriate constant c.*

More precisely, $S_2(\mathcal{H}_{m,d}) \leq m^d \log^d m$ for $m \geq 3$.

Proof. Let S be a k-TC-spanner for the line $\mathcal{H}_{m,1}$. By Lemma 4.1, $S \circ \cdots \circ S$, where the strong graph product is applied d times, is a k-TC-spanner for the directed grid $\mathcal{H}_{m,d}$. By definition of the strong graph product, the number of edges in the resulting spanner is $(|E(S)| + m)^d - m^d$. Since the number of edges in the spanner, $|E(S)|$, is at least m, the main statement follows.

The more precise statement for $k = 2$ follows from Claim 4.3 below which gives a more careful analysis of the size of the sparsest 2-TC-spanner of the line: namely, $S_2(\mathcal{H}_{m,1}) \leq m \log m - m$ for $m \geq 3$. □

Claim 4.3. *For all $m \geq 3$, the directed line $\mathcal{H}_{m,1}$ has a 2-TC-spanner with at most $m \log m - m$ edges.*

Proof. Construct graph S on vertex set $[m]$ recursively. First, define the middle node $v_{mid} = \lceil \frac{m}{2} \rceil$. Add edges (v, v_{mid}) for all nodes $v < v_{mid}$ and edges (v_{mid}, v) for all nodes $v > v_{mid}$. Then recurse on the two line segments resulting from removing v_{mid} from the current line. Proceed until each line segment contains exactly one node. This construction is implicit in, e.g., [12].

S is a 2-TC-spanner for the line $\mathcal{H}_{m,1}$, since every pair of nodes $u, v \in [m]$ is connected by a path of length at most 2 via a middle node. This happens in the stage of the recursion where u and v are separated into different line segments, or one of these two nodes is removed.

There are $t = \lfloor \log m \rfloor$ stages of the recursion, and in each stage $i \in [t]$ each node that is not removed by the end of the this stage connects to the middle node in its current line segment. Since 2^{i-1} nodes are removed in the ith stage, exactly $m - (2^i - 1)$ edges are added in that stage. Thus, the total number of edges in S is $m \cdot t - (2^{t+1} - t - 2) \leq m \log m - m$. The last inequality holds for $m \geq 3$. $\square$

4.2 Lower Bound

In this section, we show the lower bound on $S_2(\mathcal{H}_{m,d})$ stated in Theorem 1.5. We first treat the special case of this lower bound for $d = 2$, since it already contains most of the difficulty of the larger dimensional case. The extension to arbitrary dimension is presented in the subsequent section.

4.2.1 Lower Bound for $d = 2$

In this section, we prove a lower bound on the size of a 2-TC-spanner of the 2-dimensional directed grid, stated in Theorem 4.4. This is a special case of the lower bound in Theorem 1.5.

Theorem 4.4. *Any 2-TC-spanner of the 2-dimensional grid $\mathcal{H}_{m,2}$ must have $\Omega\left(\frac{m^2 \log^2 m}{\text{loglog}\, m}\right)$ edges.*

One way to prove the $\Omega(m \log m)$ lower bound on the size of a 2-TC-spanner for the directed line $\mathcal{H}_{m,1}$, stated in Lemma 1.7, is to observe that at least $\lfloor \frac{m}{2} \rfloor$ edges are cut when the line is halved: namely, at least one per vertex pair $(v, m - v + 1)$ for all $v \in \left[\lfloor \frac{m}{2} \rfloor\right]$. Continuing to halve the line recursively, we obtain the desired bound.

A natural extension of this approach to proving a lower for the grid is to recursively halve the grid along both dimensions, hoping that each such operation on an $m \times m$ grid cuts $\Omega(m^2 \log m)$ edges. This would imply that the size $S(m)$ of a 2-TC-spanner of the $m \times m$ grid satisfies the recurrence $S(m) = 4S(m/2) + \Omega(m^2 \log m)$; that is, $S(m) = \Omega(m^2 \log^2 m)$, matching the upper bound in Theorem 1.5.

An immediate problem with this approach is that in some 2-TC-spanners of the grid only $O(m^2)$ edges connect vertices in different quarters. One example of such a 2-TC-spanner is the graph containing the transitive closure of each quarter and only at most $3m^2$ edges crossing from one quarter to another: namely, for each node u and each quarter q with vertices comparable to u, this graph contains an edge (u, v_q), where v_q is the smallest node in q comparable to u.

The TC-spanner in the example above is not optimal because it has too many edges inside the quarters. The first step in our proof of Theorem 4.4 is understanding the tradeoff between the number of edges *crossing* the cut and the number of edges *internal* to the subgrids, resulting from halving the grid along some dimension. The simplest manifestation of this tradeoff occurs when a $2 \times m$ grid is halved into two lines. (In the case of one line, there is no trade off: the $\Omega(m)$ bound on the number of crossing edges holds even if each half-line contains all edges of its transitive closure.) Lemma 4.5 formulates the tradeoff for the two-line case, while taking into account only edges needed to connect comparable vertices on different lines by paths of length at most 2:

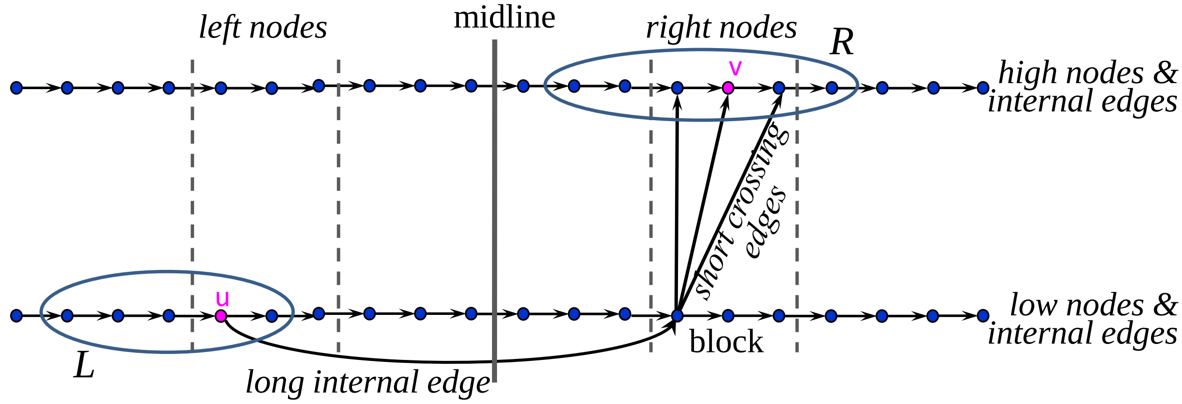

Figure 1: Illustration of the first stage in the proof of Lemma 4.5.

Lemma 4.5 (Two-Lines Lemma)**.** *Let U be a graph with vertex set $[2] \times [m]$ that contains a path of length at most 2 from u to v for every $u \in \{1\} \times [m]$ and $v \in \{2\} \times [m]$, where $u \preceq v$. An edge (u, v) in U is called* internal *if $u_1 = v_1$, and* crossing *otherwise. If U contains at most $\frac{m \log^2 m}{32}$ internal edges, it must contain at least $\frac{m \log m}{16 \log \log m}$ crossing edges.*

Note that if the number of internal edges is unrestricted, a 2-TC-spanner of $\mathcal{H}_{m,2}$ may have only m crossing edges.

Proof. The proof proceeds in $\frac{\log m}{2 \log \log m}$ stages dealing with pairwise disjoint sets of crossing edges. In each stage, we show that U contains at least $\frac{m}{8}$ crossing edges in the prescribed set.

In the first stage, divide U into $\log^2 m$ blocks, each of length $\frac{m}{\log^2 m}$: namely, a node (v_1, v_2) is in block i if $v_2 \in \left[\frac{(i-1) \cdot m}{\log^2 m} + 1, \frac{i \cdot m}{\log^2 m}\right]$. Call an edge *long* if it starts and ends in different blocks, and *short* otherwise. Assume, for contradiction, that U contains fewer than $\frac{m}{8}$ long crossing edges.

Call a node (v_1, v_2) *low* if $v_1 = 1$ (*high* if $v_1 = 2$), and *left* if $v_2 \in \left[\frac{m}{2}\right]$ (*right* otherwise). Also, call an edge (u, v) *low-internal* if $u_1 = v_1 = 1$ and *high-internal* if $u_1 = v_1 = 2$. Let L be the set of low left nodes that are not incident to long crossing edges. Similarly, let R be the set of high right nodes that are not incident to long crossing edges. Since there are fewer than $\frac{m}{8}$ long crossing edges, $|L| > \frac{m}{4}$ and $|R| > \frac{m}{4}$.

A node $u \in L$ can connect to a node $v \in R$ via a path of length at most 2 only by using a long internal edge. Observe that each long low-internal edge can be used by at most $\frac{m}{\log^2 m}$ such pairs (u, v): one low node u and high nodes v from one block. This is illustrated in Figure 1. Analogously, every long high-internal edge can be used by at most $\frac{m}{\log^2 m}$ such pairs. Since $|L| \cdot |R| > \frac{m^2}{16}$ pairs in $L \times R$ connect via paths of length at most 2, graph U contains more than $\frac{m^2}{16} \cdot \frac{\log^2 m}{m} = \frac{m \log^2 m}{16}$ long internal edges, which is a contradiction.

In each subsequent stage, call blocks used in the previous stage *megablocks*, and denote their length by B. Subdivide each megablock into $\log^2 m$ blocks of equal size. Call an edge *long* if it starts and ends in different blocks, but stays within one megablock. Assume, for contradiction, that U contains fewer than $\frac{m}{8}$ long crossing edges.

Call a node (v_1, v_2) *left* if it is in the left half of its megablock, that is, if $v_2 \leq \frac{\ell + r}{2}$ whenever (v_1, v_2) is in a megablock $[2] \times \{\ell, \ldots, r\}$. (Call it *right* otherwise). Consider megablocks containing fewer than $\frac{B}{4}$ long crossing edges each. By an averaging argument, at least $\frac{m}{2B}$ megablocks are of this type. (Recall that there are $\frac{m}{B}$ megablocks in total). Within each such megablock more than $\frac{B}{4}$ low left nodes and more than

$\frac{B}{4}$ high right nodes have no incident long crossing edges. By the argument from the first stage, each such megablock contributes more than $\frac{B^2}{16b}$ long internal edges, where $b = \frac{B}{\log^2 m}$ is the size of the blocks. Hence there must be more than $\frac{B^2}{16b} \cdot \frac{m}{2B} = \frac{m \log^2 m}{32}$ long internal edges, which is a contradiction to the fact that U contains at most $\frac{m \log^2 m}{32}$ internal edges.

We proceed to the next stage until each block is of length 1. Therefore, the number of stages, t, satisfies $\frac{m}{\log^{2t} m} = 1$. That is, $t = \frac{\log m}{2 \log \log m}$, and each stage contributes $\frac{m}{8}$ new crossing edges, as desired. □

Next we generalize Lemma 4.5 to understand the tradeoff between the number of internal edges and crossing edges resulting from halving a 2-TC-spanner of a $2\ell \times m$ grid with the usual partial order.

Lemma 4.6. *Let S be a 2-TC-spanner of the directed $[2\ell] \times [m]$ grid. An edge (u, v) in S is called* internal *if $u_1, v_1 \in [\ell]$ or $u_1, v_1 \in \{\ell + 1, \ldots, 2\ell\}$, and* crossing *otherwise. If S contains at most $\frac{\ell m \log^2 m}{64}$ internal edges, it must contain at least $\frac{\ell m \log m}{32 \log \log m}$ crossing edges.*

Proof. For each $i \in [\ell]$, we match the lines $\{i\} \times [m]$ and $\{2\ell - i + 1\} \times [m]$. Observe that a path of length at most 2 between the matched lines cannot use any edges with both endpoints in $\{i + 1, \ldots, 2\ell - i\} \times [m]$. We modify S to ensure that there are no edges with only one endpoint in $\{i + 1, \ldots, 2\ell - i\} \times [m]$ for all $i \in [\ell]$, and then apply Lemma 4.5 to the matched pairs of lines.

Call the $[\ell] \times [m]$ subgrid and all vertices and edges it contains *low*, and the remaining $\{\ell+1, \ldots, 2\ell\} \times [m]$ subgrid and its vertices and edges *high*. Transform S into S' as follows: change each low internal edge (u, v) to $(u, (u_1, v_2))$, change each high internal edge (u, v) to $((v_1, u_2), v)$, and finally change each crossing edge $((i_1, j_1), (2\ell - i_2 + 1, j_2))$ to $((i, j_1), (2\ell - i + 1, j_2))$, where $i = \min(i_1, i_2)$. Intuitively, we are projecting the edges in S to be fully contained in one of the matched pairs of lines, while preserving whether the edge is internal or crossing. Crossing edges are projected onto the outer matched pair of lines chosen from the two pairs that contain the endpoints of a given edge.

Clearly, S' contains at most the number of internal (crossing) edges as S. Observe that S' contains a path of length at most 2 from u to v for every comparable pair (u, v) where u is low, v is high, and u and v belong to the same pair of matched lines. Indeed, since S is a 2-TC-spanner, it contains either the edge (u, v) or a path (u, w, v). In the first case, S' also contains (u, v). In the second case, if (u, w) is a crossing edge S' contains $(u, (v_1, w_2), v)$, and if (u, w) is an internal edge S' contains $(u, (u_1, w_2), v)$. As claimed, each edge in S' belongs to one of the matched pairs of lines.

Finally, we apply Lemma 4.5. If S contains at most $\frac{\ell m \log^2 m}{64}$ internal edges, then so does S', and so at least half $\left(\text{i.e., } \frac{\ell}{2}\right)$ of the matched line pairs each contain at most $\frac{m \log^2 m}{32}$ internal edges. By Lemma 4.5, each of these pairs contributes at least $\frac{m \log m}{16 \log \log m}$ crossing edges. Thus S' must contain at least $\frac{\ell m \log m}{32 \log \log m}$ crossing edges. Since S contains as many crossing edges as S', the lemma follows. □

Now we prove Theorem 4.4 by recursively halving $\mathcal{H}_{m,2}$ along the horizontal dimension. Some resulting $\ell \times m$ subgrids may violate Lemma 4.6, but we can guarantee that the lemma holds for a constant fraction of the recursive steps for which $\ell \geq \sqrt{m}$. This is sufficient for obtaining the lower bound in the theorem.

Proof of Theorem 4.4. Assume m is a power of 2 for simplicity.
For each step $i \in \{1, \ldots, \frac{1}{2} \log m\}$, partition $\mathcal{H}_{m,2}$ into the following 2^{i-1} equal-sized subgrids: $\{1, \ldots, l_i\} \times [m]$, $\{l_i + 1, \ldots, 2l_i\} \times [m]$, $\ldots$, $\{m - l_i + 1, \ldots, m\} \times [m]$ where $l_i = m/2^{i-1}$. For each of these subgrids, define internal and crossing edges as in Lemma 4.6. Now, suppose that there exists a step i such that at least half of the 2^{i-1} subgrids have $> \frac{l_i m \log^2 m}{64}$ internal edges. Since at a fixed i, the subgrids are disjoint, there are $2^{i-1}\Omega(l_i m \log^2 m) = \Omega(m^2 \log^2 m)$ edges in S, proving the theorem. On the other hand, suppose that for every $i \in \{1, \ldots, \frac{1}{2} \log m\}$, at least half of the 2^{i-1} subgrids have $\leq \frac{l_i m \log^2 m}{64}$

internal edges. Then, applying Lemma 4.6, the number of crossing edges in those subgrids is $\geq \frac{l_i m \log m}{32 \log\log m}$. Counting over all steps i and for all appropriate subgrids from those steps, the number of edges in S is bounded by $\Omega\left(m^2 \log m \frac{\log m}{\log\log m}\right) = \Omega\left(m^2 \frac{\log^2 m}{\log\log m}\right)$. $\square$

4.2.2 Lower Bound for general d

In this section, we extend the above proof to establish lower bounds on $S_2(\mathcal{H}_{m,d})$ for arbitrary $d \geq 2$. The main technical result is a tradeoff lemma between internal and crossing edges with respect to two $(d-1)$-dimensional hyperplanes. An important part of the generalization is the appropriate definition of the notions of blocks and megablocks, so that the iterative argument in the proof of Lemma 4.5 applies in the high-dimensional setting.

The following theorem implies the lower bound expression in Theorem 1.5:

Theorem 4.7. *Any 2-TC-spanner of $\mathcal{H}_{m,d}$ has at least $\frac{m^d}{32} \frac{\log^d m}{(2d \log\log m)^{d-1}}$ edges.*

The main ingredient in the proof is the Two-Hyperplanes Lemma, an analogue of the Two-Lines Lemma (Lemma 4.5) for d dimensions. The main difficulty in extending the proof of the Two-Lines lemma to work for two hyperplanes is in generalizing the definitions of blocks and megablocks, so that, on one hand, each stage in the proof contributes a substantial number of crossing edges and, on the other hand, the crossing edges contributed in separate stages are pairwise disjoint.

Lemma 4.8 (Two-Hyperplanes Lemma). *Let U be a graph with vertex set $[2] \times [m]^{d-1}$ that contains a path of length at most 2 from u to v for every $u \in \{1\} \times [m]^{d-1}$ and $v \in \{2\} \times [m]^{d-1}$, where $u \preceq v$. As in Lemma 4.5, an edge (u, v) in U is called* internal *if $u_1 = v_1$, and* crossing *otherwise. Then, if U contains less than $\frac{m^{d-1} \log^d m}{(d-1)2^{2d+3}}$ internal edges, it must contain $\geq \frac{m^{d-1}}{8} \left(\frac{\log m}{2d \log\log m}\right)^{d-1}$ crossing edges.*

Proof. As for Lemma 4.5, the proof proceeds in several stages. The stages are indexed by $(d-1)$-tuples $\boldsymbol{i}$ in $\{0, 1, \ldots, \frac{\log m}{d \log\log m} - 1\}^{d-1}$. Then, the number of stages is $\left(\frac{\log m}{d \log\log m}\right)^{d-1}$. We show below that each stage contributes at least $\frac{m^{d-1}}{2^{d+2}}$ separate edges to the set of crossing edges, thus proving our lemma.

As in the proof of Lemma 4.5, at each stage vertices are partitioned into megablocks and blocks. In stage $\boldsymbol{i} = (i_1, \ldots, i_{d-1})$, we partition U into $(\log m)^{d(i_1 + \cdots + i_{d-1})}$ equal-sized *megablocks* indexed by $\boldsymbol{b} = (b_1, \ldots, b_{d-1})$, where $b_j \in [\log^{d \cdot i_j} m]$ for all $j \in [d-1]$.

A vertex v is in a megablock $\boldsymbol{b}$ if $v_{j+1} \in \left[(b_j - 1)\frac{m}{\log^{d i_j} m} + 1, b_j \frac{m}{\log^{d i_j} m}\right]$ for each $j \in [d-1]$. So, initially when $\boldsymbol{i} = \vec{0}$, there is only one megablock, and each time $\boldsymbol{i}$ increases by 1 in one coordinate, the volume of the megablocks shrinks by a factor of $\log^d m$.

Each megablock $\boldsymbol{b}$ is further partitioned into $(\log m)^{d(d-1)}$ equal-sized *blocks* indexed by $\boldsymbol{c} \in [\log^d m]^{d-1}$.

A vertex v in a megablock $\boldsymbol{b}$ lies in block $\boldsymbol{c}$ if $(v - \boldsymbol{b}_{min})_{j+1} \in \left[(c_j - 1)\frac{\ell_j}{\log^d m} + 1, c_j \frac{\ell_j}{\log^d m}\right]$ for each $j \in [d-1]$, where $\boldsymbol{b}_{min}$ denotes the smallest vertex contained in megablock $\boldsymbol{b}$ and ℓ_j denotes the length of $\boldsymbol{b}$ in the the j'th dimension. Note that vertices $(1, v_2, \ldots, v_d)$ and $(2, v_2, \ldots, v_d)$ belong to the same (mega)block. At the last stage, each block contains only two vertices (differing by the first coordinate).

Next, we specify the set of crossing edges contributed at each stage. A crossing edge (u, v) in U is said to be *long* in stage $\boldsymbol{i}$ if:

(i) u and v lie in the same megablock, and

(ii) If u lies in block $(c_1, \ldots, c_{d-1})$ and v lies in block $(c'_1, \ldots, c'_{d-1})$, then $c_j < c'_j$ for all $j \in [d-1]$.

We claim that if $\boldsymbol{i} \neq \boldsymbol{i'}$, the sets of long crossing edges in stages $\boldsymbol{i}$ and $\boldsymbol{i'}$ are disjoint. To see this, let j be an index such that $i_j \neq i'_j$; suppose without loss of generality that $i_j < i'_j$. Then, the length of the megablocks in the j'th dimension for stage $\boldsymbol{i'}$ is at most the length of the blocks in the j'th dimension for stage $\boldsymbol{i}$. Hence, condition (ii) above implies that long crossing edges in stage $\boldsymbol{i}$ must have endpoints in different megablocks of stage $\boldsymbol{i'}$, and so violate condition (i) for being a long crossing edge in stage $\boldsymbol{i'}$.

It remains to show that every stage contributes at least $\frac{m^{d-1}}{2^{d+2}}$ long crossing edges. For the sake of contradiction, suppose that the number of long crossing edges at some stage $\boldsymbol{i}$ is $< \frac{m^{d-1}}{2^{d+2}}$. Let $B = m^{d-1}/(\log m)^{d(i_1+\cdots+i_{d-1})}$ be the volume of the megablocks restricted to one of the two hyperplanes. By an averaging argument, at least $\frac{m^{d-1}}{2B}$ megablocks contain $< \frac{B}{2^{d+1}}$ long crossing edges (otherwise, there would be at least $\frac{m^{d-1}}{2^{d+2}}$ long crossing edges). But we show next that if a megablock contains $< \frac{B}{2^{d+1}}$ long crossing edges, then there are $\geq \frac{B\log^d m}{(d-1)2^{2d+2}}$ internal edges with both endpoints inside the megablock. This would imply that the total number of internal edges is $\geq \frac{m^{d-1}}{2B} \cdot \frac{B\log^d m}{(d-1)2^{2d+2}} = \frac{m^{d-1}\log^d m}{(d-1)2^{2d+3}}$, a contradiction.

Suppose then that a megablock contains $< \frac{B}{2^{d+1}}$ long crossing edges. Let Low be the set of vertices in the megablock with each coordinate at most the average value of that coordinate in the megablock, and $High$ the set of vertices with each coordinate greater than the average value of that coordinate. Then $|Low| \geq \frac{B}{2^d}, |High| \geq \frac{B}{2^d}$, and each vertex in Low is comparable to each vertex in $High$. By the bound on the number of long crossing edges, there must exist a set L of at least $\frac{B}{2^{d+1}}$ vertices in Low not incident to any long crossing edge, and a set R of at least $\frac{B}{2^{d+1}}$ vertices in $High$ not incident to any long crossing edges. L lies in the lower hyperplane, R in the upper hyperplane, and each vertex in L is comparable to each vertex in R. Call a crossing edge *short* if it satisfies condition (i), but violates condition (ii) above. A path in U of length at most 2 from a vertex in L to a vertex in R must consist of one internal edge and one short crossing edge. The number of short crossing edges incident to a given vertex v is at most $(d-1)\frac{B}{\log^d m}$, by counting, for each of the $d-1$ block indices, the number of vertices in the megablock that share the value of that block index with v. So, each internal edge helps connect at most $(d-1)\frac{B}{\log^d m}$ pairs of vertices. Since $\frac{B^2}{2^{2d+2}}$ pairs of vertices need to be connected by a path, there must exist at least $\frac{B^2}{2^{2d+2}} \cdot \frac{\log^d m}{(d-1)B} = \frac{B\log^d m}{(d-1)2^{2d+2}}$ internal edges. □

The analogue of Lemma 4.6 in d dimensions (Lemma 4.9) and the rest of the proof of Theorem 4.7 are straightforward generalizations of the 2-dimensional case.

Lemma 4.9. *Let S be a 2-TC-spanner of the directed $[2\ell]\times[m]^{d-1}$ grid. An edge (u,v) in S is called* internal *if $u_1, v_1 \in [\ell]$ or $u_1, v_1 \in \{\ell+1,\ldots,2\ell\}$, and* crossing *otherwise. If S contains less than $\frac{\ell m^{d-1}\log^d m}{(d-1)2^{2d+3}}$ internal edges, it must contain at least $\geq \ell\frac{m^{d-1}}{8}\left(\frac{\log m}{2d\log\log m}\right)^{d-1}$ crossing edges.*

Proof sketch. We can generalize the proof of Lemma 4.6 in a straightforward way. For each $i \in [\ell]$, instead of matching the lines, we match the hyperplanes $\{i\} \times [m]^{d-1}$ and $\{2\ell - i + 1\} \times [m]^{d-1}$. □

Proof of Theorem 4.7. Assume m is a power of 2 for simplicity.
For each step $i \in \{1,\ldots,\frac{1}{2}\log m\}$, partition $\mathcal{H}_{m,d}$ into the following 2^{i-1} equal-sized subgrids: $\{1,\ldots,l_i\}$ $\times[m]^{d-1}$, $\{l_i+1,\ldots,2l_i\}\times[m]^{d-1},\ldots,\{m-l_i+1,\ldots,m\}\times[m]^d$ where $l_i = m/2^{i-1}$. For each of these subgrids, define internal and crossing edges as in Lemma 4.9. Now, suppose that there exists a step i such that at least half of the 2^{i-1} subgrids have $\geq \frac{l_i m^{d-1}\log^d m}{(d-1)2^{2d+3}}$ internal edges. Since at a fixed i, the subgrids are disjoint, there are at least $2^{i-2}\frac{l_i m^{d-1}\log^d m}{(d-1)2^{2d+3}} = \frac{m^d\log^d m}{(d-1)2^{2d+4}}$ edges in S, which is enough to prove the theorem. On the other hand, suppose that for every $i \in \{1,\ldots,\frac{1}{2}\log m\}$, at least half of the 2^{i-1} subgrids have

$< \frac{l_i m^{d-1} \log^d m}{(d-1)2^{2d+3}}$ internal edges. Then, applying Lemma 4.9, the number of crossing edges in those subgrids is $\geq \frac{l_i m^{d-1}}{8} \left(\frac{\log m}{2d \log\log m}\right)^{d-1}$. Counting over all steps i and for all appropriate subgrids from those steps, the number of edges in S is lower-bounded by $\frac{\log m}{2} \cdot 2^{i-2} \cdot \frac{l_i m^{d-1}}{8} \left(\frac{\log m}{2d \log\log m}\right)^{d-1} = \frac{m^d}{32} \frac{\log^d m}{(2d \log\log m)^{d-1}}$. $\square$

5 2-TC-Spanners of the Hypercube

In this section we prove Theorem 1.6, namely, we analyze the size of the sparsest 2-TC-spanner of the d-dimensional hypercube $\mathcal{H}_d$. Lemma 5.1 presents the upper bound on $S_2(\mathcal{H}_d)$. Lemma 5.3 presents the lower bound. The upper and lower bounds differ only by a factor of $O(d^3)$, and are dominated by the same combinatorial expression. A numerical approximation to this expression is given in Lemma 5.5. Remark 5.1 at the end of the section explains why our randomized construction in Lemma 5.1 yields a 2-TC-spanner of $\mathcal{H}_d$ of size within $O(d^2)$ of the optimal.

5.1 Upper Bound

Lemma 5.1. *There is a 2-TC-spanner of $\mathcal{H}_d$ with $O\left(d^3 \max_{i,j:i<j} \min_{k:i\leq k\leq j} \frac{\binom{d}{k}}{\binom{j-i}{k-i}} \max\left\{\binom{k}{i}, \binom{d-k}{d-j}\right\}\right)$ edges.*

Proof. Consider the following probabilistic construction that connects all comparable vertices at levels i and j of $\mathcal{H}_d$ by paths of length at most 2:

Given levels $i, j \in \{0, 1, ..., d\}$, $i < j$,

1. Initialize the set $E_{i,j}$ to $\emptyset$.
2. Let $k_{i,j} = \underset{k:i\leq k\leq j}{\text{argmin}} \left(\frac{\binom{d}{k}}{\binom{j-i}{k-i}} \max\left\{\binom{k}{i}, \binom{d-k}{d-j}\right\}\right)$.
3. Let $S_{i,j}$ be a set of $3d\frac{\binom{d}{k}}{\binom{j-i}{k-i}}$ vertices chosen uniformly at random from the set of $\binom{d}{k}$ vertices that are in weight level $k = k_{i,j}$.
4. For each vertex $v \in S_{i,j}$, set $E_{i,j}$ to $E_{i,j} \cup \{(x, v) : |x| = i \wedge x \prec v\} \cup \{(v, y) : |y| = j \wedge v \prec y\}$. That is, connect v to all comparable vertices in levels i and j.
5. Output $E_{i,j}$.

Claim 5.2. *For all $0 \leq i < j \leq d$, with probability at least $\frac{1}{2}$, $E_{i,j}$ contains a path of length at most 2 between any pair of vertices (x, y) such that $x \prec y$, $|x| = i$, and $|y| = j$.*

Proof. Consider any particular pair of vertices (x, y) such that $x \prec y$, $|x| = i$, and $|y| = j$. The number of vertices in level k that are greater than x and less than y is exactly $\binom{j-i}{k-i}$. So, the probability that $S_{i,j}$ does not contain such a vertex is: $\left(1 - \binom{j-i}{k-i}/\binom{d}{k}\right)^{3d\frac{\binom{d}{k}}{\binom{j-i}{k-i}}} \leq e^{-3d}$. The number of comparable pairs (x, y) is $\binom{d}{i}\binom{d-i}{d-j}$. So, by the union bound, the probability that there exists an (x, y) such that no vertex $v \in S_{i,j}$ satisfies $x \prec v \prec y$ is at most $\binom{d}{i}\binom{d-i}{d-j}e^{-3d} \leq 2^{2d}e^{-3d} < \frac{1}{2}$. $\square$

So, for every i and j, there exists a choice of $S_{i,j}$ such that comparable pairs from the two weight levels are connected by a path of length at most 2. Let $E^*_{i,j}$ be the set of edges returned by the algorithm when this $S_{i,j}$ is chosen. We set $E = \bigcup_{0\le i<j\le d} E^*_{i,j}$. By Claim 5.2, $(\{0,1\}^d, E)$ is a 2-TC-spanner of $\mathcal{H}_d$.

Now, we show that the size of E is as claimed in the lemma statement. The main observation is that in step (4), for any specific $v \in S_{i,j}$, $|\{(x,v) : |x| = i \wedge x \prec v\} \cup \{(v,y) : |y| = j \wedge v \prec y\}|$ is exactly $\binom{k_{i,j}}{i} + \binom{d-k_{i,j}}{d-j}$. Therefore, for all $0 \le i < j \le d$,

$$|E^*_{i,j}| \le 3d \min_{k: i\le k\le j} \frac{\binom{d}{k}}{\binom{j-i}{k-i}} \left(\binom{k}{i} + \binom{d-k}{d-j}\right) \le 6d \min_{k: i\le k\le j} \frac{\binom{d}{k}}{\binom{j-i}{k-i}} \max\left\{\binom{k}{i}, \binom{d-k}{d-j}\right\}.$$

Since $|E| = \sum_{0\le i<j\le d} |E^*_{i,j}|$, where the sum has $O(d^2)$ terms, the claimed bound follows. □

5.2 Lower Bound

Lemma 5.3. *Any 2-TC-spanner of $\mathcal{H}_d$ has $\Omega\left(\max_{i,j: i<j} \min_{k: i\le k\le j} \frac{\binom{d}{k}}{\binom{j-i}{k-i}} \max\left\{\binom{k}{i}, \binom{d-k}{d-j}\right\}\right)$ edges.*

Proof. Let S be a 2-TC-spanner for $\mathcal{H}_d$. We will count the edges in S that occur on paths connecting two particular weight levels of $\mathcal{H}_d$. Let $P_{i,j}$ be the pairs $\{(v_1, v_2) : |v_1| = i, |v_2| = j, v_1 \prec v_2\}$. We will lower bound $e^*_{i,j}$, the number of edges in the paths of length at most 2 in S that connect the pairs $P_{i,j}$. Let $e_{k,\ell}$ denote the number of edges in S that connect vertices in level k to vertices in level ℓ. Then $e^*_{i,j} = e_{i,j} + \sum_{k=i+1}^{j-1} (e_{i,k} + e_{k,j})$.

We say that a vertex v *covers* a pair of vertices (v_1, v_2) if S contains the edges (v_1, v) and (v, v_2) or, for the special case $v = v_1$, if S contains (v_1, v_2). Let $V^{(k)}_{i,j}$ be the set of vertices of weight k that cover pairs in $P_{i,j}$. Let α_k be the fraction of pairs in $P_{i,j}$ that are covered by a vertex in $V^{(k)}_{i,j}$. Since each pair in $P_{i,j}$ must be covered by a vertex in levels i to $j-1$, $\sum_{k=i}^{j-1} \alpha_k \ge 1$.

For any vertex $v \in V^{(k)}_{i,j}$, let in_v be the number of incoming edges from vertices of weight i incident to v and let out_v be the number of outgoing edges to vertices of weight j incident to v. For each $k \in \{i+1, ..., j-1\}$, since each vertex $v \in V^{(k)}_{i,j}$ covers $in_v \cdot out_v$ pairs,

$$\sum_{v\in V^{(k)}_{i,j}} in_v \cdot out_v \ge \alpha_k |P_{i,j}| = \alpha_k \binom{d}{i}\binom{d-i}{d-j}. \tag{1}$$

We upper bound $\sum_{v\in V^{(k)}_{i,j}} in_v \cdot out_v$ as a function of $e_{i,k} + e_{k,j}$, and then use Equation (1) to lower bound $e_{i,k} + e_{k,j}$.

For all $k \in \{i+1, ..., j-1\}$, variables in_v and out_v satisfy the following constraints:

$$\sum_{v\in V^{(k)}_{i,j}} in_v \le e_{i,k} + e_{k,j}, \quad \sum_{v\in V^{(k)}_{i,j}} out_v \le e_{i,k} + e_{k,j}\ .$$

$$in_v \le \binom{k}{i} \forall v \in V^{(k)}_{i,j}, \quad out_v \le \binom{d-k}{d-j} \forall v \in V^{(k)}_{i,j}\ .$$

The last two constraints hold because in_v and out_v count the number of edges to a vertex of weight k from from vertices of weight i and from a vertex of weight k to vertices of weight j, respectively. Using these bounds we obtain

$$\sum_{v\in V^{(k)}_{i,j}} in_v \cdot out_v \le \sum_{v\in V^{(k)}_{i,j}} \binom{k}{i} \cdot out_v = \binom{k}{i} \cdot \sum_{v\in V^{(k)}_{i,j}} out_v \le \binom{k}{i} \cdot (e_{i,k} + e_{k,j}).$$

Similarly, $\sum_{v\in V_{i,j}^{(k)}} in_v \cdot out_v \leq \binom{d-k}{d-j} \cdot (e_{i,k}+e_{k,j})$. Therefore, for all $k \in \{i+1,...,j-1\}$:

$$\sum_{v\in V_{i,j}^{(k)}} in_v \cdot out_v \leq (e_{i,k}+e_{k,j}) \min\left\{\binom{k}{i}, \binom{d-k}{d-j}\right\}.$$

Let $s_{i,k,j} = \frac{\binom{d}{i}\binom{d-i}{d-j}}{\min\left\{\binom{k}{i},\binom{d-k}{d-j}\right\}}$. From Equation (1), $e_{i,k}+e_{k,j} \geq \alpha_k s_{i,k,j}$ for all $k \in \{i+1,...,j-1\}$. Therefore,

$$e^*_{i,j} = e_{i,j} + \sum_{k=i+1}^{j-1}(e_{i,k}+e_{k,j}) \geq \alpha_i \binom{d}{i}\binom{d-i}{d-j} + \sum_{k=i+1}^{j-1} \alpha_k s_{i,k,j} \geq \sum_{k=i}^{j-1} \alpha_k s_{i,k,j} \geq \min_{k:i\leq k\leq j} s_{i,k,j}$$

Since this holds for arbitrary i and j, the number of edges in the 2-TC-spanner

$$|S| \geq \max_{i,j:i<j} \min_{k:i\leq k\leq j} s_{i,k,j}.$$

Finally, a simple algebraic manipulation finishes the proof.

Claim 5.4. $s_{i,k,j} = \frac{\binom{d}{k}}{\binom{j-i}{k-i}} \max\left\{\binom{k}{i},\binom{d-k}{d-j}\right\}$.

Proof. Take the ratio of the two sides:

$$\frac{s_{i,k,j}}{\frac{\binom{d}{k}}{\binom{j-i}{k-i}}\max\left\{\binom{k}{i},\binom{d-k}{d-j}\right\}} = \frac{\binom{d}{i}\binom{d-i}{d-j}\binom{j-i}{k-i}}{\binom{d}{k}\binom{k}{i}\binom{d-k}{d-j}} = \frac{\binom{d}{i}\binom{d-i}{j-i}\binom{j-i}{k-i}}{\binom{d}{k}\binom{k}{i}\binom{d-k}{j-k}} = 1.$$

The first equality follows from the fact that $\max(x,y)\cdot\min(x,y) = x\cdot y$. The last equality can be proved either by expanding the binomial coefficients into factorials, or by realizing that both $\binom{d}{i}\binom{d-i}{j-i}\binom{j-i}{k-i}$ and $\binom{d}{k}\binom{k}{i}\binom{d-k}{j-k}$ count the number of ways i red balls, $j-k$ blue balls, and $k-i$ green balls can be placed into d slots, each of which can hold one ball at most. This completes the proof of the claim. □

This completes the proof of the lemma. □

The following lemma gives a handle on the expression capturing the size of a 2-TC-Spanner.

Lemma 5.5. *Let* $s = \max_{i,j:i<j} \min_{k:i\leq k\leq j} \frac{\binom{d}{k}}{\binom{j-i}{k-i}} \max\left\{\binom{k}{i},\binom{d-k}{d-j}\right\}$. *Then* $s = 2^{cd}$*, where* $c \approx 1.1620$.

Proof. We use the fact that $\binom{n}{cn} = 2^{(H_b(c)-o_n(1))n}$, where "$o_n(1)$" is a function of n that tends to zero as n tends to infinity, and $H_b(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. Substituting $i = \alpha d, j = \beta d$ and $k = \gamma d$ in the resulting expression for s, and taking the logarithm of both sides, we get

$$\log_2 s = \max_{0\leq\alpha<\beta\leq 1} \min_{\alpha\leq\gamma\leq\beta} \left[H_b(\gamma) - H_b\left(\frac{\gamma-\alpha}{\beta-\alpha}\right)(\beta-\alpha) + \max\left(H_b\left(\frac{\alpha}{\gamma}\right)\gamma, H_b\left(\frac{1-\beta}{1-\gamma}\right)(1-\gamma)\right)\right] d$$

In other words, $\log_2 s = cd$ where c is a constant. We can check numerically that $c \approx 1.1620$. □

Remark 5.1. We note that if the first maximum in the expression for s is replaced with the sum then Lemma 5.1 holds for $O(d\cdot s)$ instead of $O(d^3\cdot s)$ while Lemma 5.3 holds for $\Omega(d/s)$ instead of $\Omega(s)$. The proofs of these modified statements are similar. (We do not have an analogue of Lemma 5.5 for the modified expression for s.) Observe that the modified bounds differ by a factor of $O(d^2)$ instead of $O(d^3)$. This demonstrates that our randomized construction yields a 2-TC-spanner of $\mathcal{H}_d$ of size within $O(d^2)$ of the optimal. ♢

6 2-TC-Spanners of High-Dimensional Hypergrids

In this section, we generalize the arguments for the hypercube (Section 5) to the directed hypergrid, $\mathcal{H}_{m,d}$, to find the size of the sparsest 2-TC-spanner for $\mathcal{H}_{m,d}$ to within $\mathrm{poly}(d)$ factors. This result supersedes the results of Section 4 when, for instance, m is constant and d is growing. The expression we obtain can be evaluated numerically for small m using standard approximations of binomial coefficients. For example, this was done in Lemma 5.5 for the case $m = 2$.

Before stating Theorem 6.1, we introduce some notation.

Definition 6.1. *For the hypergrid $\mathcal{H}_{m,d}$, define a* level *to be a set of vertices, indexed by vector $\mathbf{i} \in [d]^m$ with $i_1 + \cdots + i_m = d$, that consists of vertices $x = (x_1, \ldots, x_d) \in [m]^d$ containing i_1 positions of value 1, i_2 positions of value 2, $\ldots$, and i_m positions of value m.*

Notice that the number of vertices in level $\mathbf{i} = (i_1, i_2, \ldots, i_m)$ is the multinomial coefficient

$$\binom{d}{\mathbf{i}} = \binom{d}{i_1, ..., i_d} = \binom{d}{i_1}\binom{d-i_1}{i_2}\binom{d-i_1-i_2}{i_3}\cdots\binom{d-\sum_{l=1}^{m-1} i_l}{i_m}.$$

Indeed, there are $\binom{d}{i_1}$ choices for the coordinates of value 1. For each such choice there are $\binom{d-i_1}{i_2}$ choices for the coordinates of value 2, and repeating this argument one obtains the above expression.

For levels $\mathbf{i}, \mathbf{j} \in [d]^m$, say $\mathbf{j}$ *majorizes* $\mathbf{i}$, denoted $\mathbf{j} \succ \mathbf{i}$, if $\mathbf{j}$ contains a vertex which is above some vertex in $\mathbf{i}$, *i.e.,* , if $\displaystyle\sum_{\ell=t}^{m} j_\ell \geq \sum_{\ell=t}^{m} i_\ell$ for all $t \in \{m, m-1, ..., 1\}$.

For $\mathbf{j} \succ \mathbf{i}$, the number of vertices y at level $\mathbf{i}$ comparable to a fixed vertex x at level $\mathbf{j}$ is $\mathcal{M}(\mathbf{i}, \mathbf{j})$:

$$\binom{j_m}{i_m}\binom{j_m + j_{m-1} - i_m}{i_{m-1}}\binom{j_m + j_{m-1} + j_{m-2} - i_m - i_{m-1}}{i_{m-2}}\cdots\binom{\sum_{l=1}^{m} j_l - \sum_{l=2}^{m} i_l}{i_1}.$$

Indeed, there are $\binom{j_m}{i_m}$ choices for the coordinates of value m in y. For each such choice, there are $\binom{j_m+j_{m-1}-i_m}{i_{m-1}}$ choices for the coordinates of value $m-1$ in y, and one can repeat this argument to obtain the claimed expression.

For $\mathbf{j} \succ \mathbf{i}$, the number of vertices y at level $\mathbf{j}$ comparable to a fixed vertex x at level $\mathbf{i}$ is

$$\mathcal{N}(\mathbf{i}, \mathbf{j}) = \frac{\mathcal{M}(\mathbf{i}, \mathbf{j})\binom{d}{\mathbf{j}}}{\binom{d}{\mathbf{i}}}.$$

Indeed, there are $\mathcal{M}(\mathbf{i}, \mathbf{j})\binom{d}{\mathbf{j}}$ comparable pairs of vertices in levels $\mathbf{i}$ and $\mathbf{j}$, and level $\mathbf{i}$ contains $\binom{d}{\mathbf{i}}$ vertices. Since, by symmetry, each vertex in $\mathbf{i}$ is comparable to the same number of vertices in level $\mathbf{j}$, we get the desired expression.

Theorem 6.1. *Let*

$$\mathcal{B}(m, d) = \max_{\mathbf{i},\mathbf{j}:\mathbf{j}\succ\mathbf{i}} \ \min_{\mathbf{k}:\mathbf{i}\prec\mathbf{k}\prec\mathbf{j}} \frac{\mathcal{M}(\mathbf{i}, \mathbf{j})\binom{d}{\mathbf{j}}}{\mathcal{M}(\mathbf{i}, \mathbf{k})\mathcal{N}(\mathbf{k}, \mathbf{j})} \max\{\mathcal{M}(\mathbf{i}, \mathbf{k}), \mathcal{N}(\mathbf{k}, \mathbf{j})\}.$$

Then the number of edges in the sparsest 2-TC-spanner of the directed hypergrid $\mathcal{H}_{m,d}$ is $O\left(d^{2m}\mathcal{B}(m, d)\right)$ and $\Omega\left(\mathcal{B}(m, d)\right)$.

The bounds stated in Theorem 6.1 are presented separately as Lemma 6.2 (upper bound) and Lemma 6.4 (lower bound) below.

6.1 Upper Bound

Lemma 6.2. *There is a 2-TC-spanner of $\mathcal{H}_{m,d}$ with*

$$O\left(d^{2m} \max_{\mathbf{i},\mathbf{j}:\mathbf{j}\succ\mathbf{i}} \; \min_{\mathbf{k}:\mathbf{i}\prec\mathbf{k}\prec\mathbf{j}} \frac{\mathcal{M}(\mathbf{i},\mathbf{j})\binom{d}{\mathbf{j}}}{\mathcal{M}(\mathbf{i},\mathbf{k})\mathcal{N}(\mathbf{k},\mathbf{j})} \max\left\{\mathcal{M}(\mathbf{i},\mathbf{k}),\mathcal{N}(\mathbf{k},\mathbf{j})\right\}\right) \text{ edges.}$$

Proof. Let $v \in \mathbf{i}$ denote that vertex v belongs to level $\mathbf{i}$. Consider the following probabilistic construction that connects comparable vertices at levels $\mathbf{i}$ and $\mathbf{j}$ of $\mathcal{H}_{m,d}$ by paths of length at most 2:

Given levels $\mathbf{i},\mathbf{j} \in [m]^d$, $\mathbf{j} \succ \mathbf{i}$,

1. Initialize the set $E_{\mathbf{i},\mathbf{j}}$ to $\emptyset$.
2. Let $\mathbf{k}_{\mathbf{i},\mathbf{j}} = \operatorname*{argmin}_{\mathbf{k}:\mathbf{i}\prec\mathbf{k}\prec\mathbf{j}} \left(\frac{\mathcal{M}(\mathbf{i},\mathbf{j})\binom{d}{\mathbf{j}}}{\mathcal{M}(\mathbf{i},\mathbf{k})\mathcal{N}(\mathbf{k},\mathbf{j})} \max\left\{\mathcal{M}(\mathbf{i},\mathbf{k}),\mathcal{N}(\mathbf{k},\mathbf{j})\right\}\right)$.
3. Let $S_{\mathbf{i},\mathbf{j}}$ be a set of $d^m \frac{\mathcal{M}(\mathbf{i},\mathbf{j})\binom{d}{\mathbf{j}}}{\mathcal{M}(\mathbf{i},\mathbf{k})\mathcal{N}(\mathbf{k},\mathbf{j})}$ vertices chosen uniformly at random from the set of $\binom{d}{\mathbf{k}}$ vertices that are in weight level $\mathbf{k} = \mathbf{k}_{\mathbf{i},\mathbf{j}}$.
4. For each vertex $v \in S_{\mathbf{i},\mathbf{j}}$, set $E_{\mathbf{i},\mathbf{j}}$ to $E_{\mathbf{i},\mathbf{j}} \cup \{(x,v) : x \in \mathbf{i} \wedge x \prec v\} \cup \{(v,y) : y \in \mathbf{j} \wedge v \prec y\}$. That is, connect v to all comparable vertices in levels $\mathbf{i}$ and $\mathbf{j}$.
5. Output $E_{\mathbf{i},\mathbf{j}}$.

Claim 6.3. *For all $\mathbf{i} \prec \mathbf{j}$, with probability at least $\frac{1}{2}$, $E_{\mathbf{i},\mathbf{j}}$ contains a path of length at most 2 between any pair of vertices (x,y) such that $x \prec y$, $x \in \mathbf{i}$, and $y \in \mathbf{j}$.*

Proof. Fix x, y with $x \prec y$, and assume $x \in \mathbf{i}$, and $y \in \mathbf{j}$. We will first show that $Pr_{v\in\mathbf{k}}[x \prec v \prec y] \geq p$, where $p = \frac{\mathcal{M}(\mathbf{i},\mathbf{k})\mathcal{N}(\mathbf{k},\mathbf{j})}{\mathcal{M}(\mathbf{i},\mathbf{j})\binom{d}{\mathbf{j}}}$.

Toward that end, notice that there are $\mathcal{M}(\mathbf{i},\mathbf{j})\binom{d}{\mathbf{j}}$ pairs of comparable vertices (u,w) with $u \in \mathbf{i}, w \in \mathbf{j}$. Each vertex in $S_{\mathbf{i},\mathbf{j}}$ connects exactly $\mathcal{M}(\mathbf{i},\mathbf{k})\mathcal{N}(\mathbf{k},\mathbf{j})$ pairs of nodes from levels $\mathbf{i}$ and $\mathbf{j}$. It is enough to show that for any such pair (u,w), the number of vertices at level $\mathbf{k}$ that are comparable to both u and v is independent of u, w, i.e., that number only depends on the levels $\mathbf{i},\mathbf{k},\mathbf{j}$ and it is therefore the same for all such pairs. To see that, for a vertex $u \in \mathbf{z}$, denote by $T_l(u)$ the set of positions of value l in u. Notice that $|T_l(u)| = z_l$. For $x \prec v \prec y$ it is the case that $T_m(x) \subseteq T_m(v) \subseteq T_m(y)$. Hence there are $\binom{j_m - i_m}{k_m - i_m}$ choices for the m-values in the vector v. Similarly, we must have $T_{m-1}(x) \subseteq T_{m-1}(v) \subseteq T_m(y) \cup T_{m-1}(y)$. Hence there are $\binom{j_m + j_{m-1} - k_m - i_{m-1}}{k_{m-1} - i_{m-1}}$ choices for the values $m-1$ in v. Repeating this process, we obtain that the number of possible v's does not depend on the particular choice of x and y.

Thus the probability that $S_{\mathbf{i},\mathbf{j}}$ does not contain such a vertex v with $x \prec v \prec y$ is $(1-p)^{d^m/p} \leq e^{-d^m}$.

The number of comparable pairs (x,y) is at most m^{2d}, and by the union bound, the probability that there exists (x,y) such that there is no $v \in S_{\mathbf{i},\mathbf{j}}$ with $x \prec v \prec y$ is at most $m^{2d}e^{-d^m} < 1/2$. $\square$

So, for every $\mathbf{i}$ and $\mathbf{j}$, there exists a choice of $S_{\mathbf{i},\mathbf{j}}$ such that comparable pairs from the two weight levels are connected by a path of length at most 2. Let $E^*_{\mathbf{i},\mathbf{j}}$ be the set of edges returned by the algorithm when this $S_{\mathbf{i},\mathbf{j}}$ is chosen. We set $E = \bigcup_{i<j} E^*_{\mathbf{i},\mathbf{j}}$. By Claim 6.3, $([m]^d, E)$ is a 2-TC-spanner of $\mathcal{H}_{m,d}$.

Now, we show that the size of E is as claimed in the lemma statement. The main observation is that in step (4), for any specific $v \in S_{\mathbf{i},\mathbf{j}}$, $|\{(x,v) : x \in \mathbf{i} \wedge x \prec v\} \cup \{(v,y) : y \in \mathbf{j} \wedge v \prec y\}|$ is exactly $\mathcal{M}(\mathbf{i},\mathbf{k}) + \mathcal{N}(\mathbf{k},\mathbf{j})$.

The claimed bound follows since $|E| = \sum_{\mathbf{j}\succ\mathbf{i}} |E^*_{\mathbf{i},\mathbf{j}}|$, where the sum has d^m terms.

$\square$

6.2 Lower Bound

Lemma 6.4. *Any 2-TC-spanner of $\mathcal{H}_{m,d}$ has at least $\Omega(\mathcal{B}(m,d))$ many edges, where $\mathcal{B}(m,d)$ is defined as in Theorem 6.1.*

Proof. Let S be a 2-TC-spanner for $\mathcal{H}_{m,d}$. We count the edges in S that occur on paths connecting two particular levels of $\mathcal{H}_{m,d}$. Let $P_{\mathbf{i},\mathbf{j}} = \{(v_1, v_2) : v_1 \in \mathbf{i}, v_2 \in \mathbf{j}, v_1 \prec v_2\}$. We will lower bound $e^*_{\mathbf{i},\mathbf{j}}$, the number of edges in the paths of length at most 2 in S, that connect the pairs $P_{\mathbf{i},\mathbf{j}}$. Notice that $|P(\mathbf{i},\mathbf{j})| = \binom{d}{\mathbf{j}}\mathcal{M}(\mathbf{i},\mathbf{j})$.

Let $e_{\mathbf{k},\boldsymbol{\ell}}$ denote the number of edges in S that connect vertices in level $\mathbf{k}$ to vertices in level $\boldsymbol{\ell}$. Then

$$e^*_{\mathbf{i},\mathbf{j}} = e_{\mathbf{i},\mathbf{j}} + \sum_{\mathbf{i}\prec\mathbf{k}\prec\mathbf{j}} (e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}). \tag{2}$$

We say that a vertex v *covers* a pair of vertices (v_1, v_2) if S contains the edges (v_1, v) and (v, v_2) or, for the special case $v = v_1$, if S contains (v_1, v_2). Let $V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}$ be the set of vertices in level $\mathbf{k}$ that cover pairs in $P_{\mathbf{i},\mathbf{j}}$. Let $\alpha_{\mathbf{k}}$ be the fraction of pairs in $P_{\mathbf{i},\mathbf{j}}$ that are covered by the vertices in $V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}$. Since each pair in $P_{\mathbf{i},\mathbf{j}}$ must be covered by a vertex in levels $\mathbf{k}$ with $\mathbf{i} \prec \mathbf{k} \prec \mathbf{j}$, we must have $\sum_{\mathbf{i}\prec\mathbf{k}\prec\mathbf{j}} \alpha_{\mathbf{k}} \geq 1$.

For any vertex $v \in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}$, let in_v be the number of incoming edges from vertices of level $\mathbf{i}$ incident to v and let out_v be the number of outgoing edges to vertices of level $\mathbf{j}$ incident to v. For each level $\mathbf{k}$ with $\mathbf{i} \prec \mathbf{k} \prec \mathbf{j}$, since each vertex $v \in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}$ covers $in_v \cdot out_v$ pairs,

$$\sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} in_v \cdot out_v \geq \alpha_{\mathbf{k}} |P_{\mathbf{i},\mathbf{j}}| \geq \alpha_{\mathbf{k}} \mathcal{M}(\mathbf{i},\mathbf{j}) \binom{d}{\mathbf{j}}. \tag{3}$$

We upper bound $\sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} in_v \cdot out_v$ as a function of $e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}$, and then use Equation (3) to lower bound $e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}$. For all $\mathbf{k}$ with $\mathbf{i} \prec \mathbf{k} \prec \mathbf{j}$, variables in_v and out_v satisfy the following constraints:

$$\sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} in_v \leq e_{\mathbf{i},\mathbf{k}} \leq e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}, \quad \sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} out_v \leq e_{\mathbf{k},\mathbf{j}} \leq e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}},$$

$$in_v \leq \mathcal{M}(\mathbf{i},\mathbf{k})\ \forall v \in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}, \quad out_v \leq \mathcal{N}(\mathbf{k},\mathbf{j})\ \forall v \in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}.$$

The last two constraints hold because in_v and out_v count the number of edges to a vertex of level $\mathbf{k}$ from vertices of level $\mathbf{i}$, and from a vertex of level $\mathbf{k}$ to vertices of level $\mathbf{j}$, respectively. Using these bounds we obtain

$$\sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} in_v \cdot out_v \leq \sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} \mathcal{M}(\mathbf{i},\mathbf{k}) \cdot out_v = \mathcal{M}(\mathbf{i},\mathbf{k}) \cdot \sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} out_v \leq \mathcal{M}(\mathbf{i},\mathbf{k}) \cdot (e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}).$$

Similarly, $\sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} in_v \cdot out_v \leq \mathcal{N}(\mathbf{k},\mathbf{j}) \cdot (e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}})$. Therefore,

$$\sum_{v\in V^{(\mathbf{k})}_{\mathbf{i},\mathbf{j}}} in_v \cdot out_v \leq (e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}) \min\{\mathcal{M}(\mathbf{i},\mathbf{k}), \mathcal{N}(\mathbf{k},\mathbf{j})\}.$$

From Equation (3), $e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}} \geq \alpha_{\mathbf{k}} \mathcal{M}(\mathbf{i},\mathbf{j}) \binom{d}{\mathbf{j}} \frac{1}{\min\{\mathcal{M}(\mathbf{i},\mathbf{k}), \mathcal{N}(\mathbf{k},\mathbf{j})\}}$ for all $\mathbf{i} \prec \mathbf{k} \prec \mathbf{j}$. Applying Equation (2) and the fact that $\sum_{\mathbf{i} \prec \mathbf{k} \prec \mathbf{j}} \alpha_{\mathbf{k}} \geq 1$, we get

$$
\begin{aligned}
e^*_{\mathbf{i},\mathbf{j}} &= e_{\mathbf{i},\mathbf{j}} + \sum_{\mathbf{i} \prec \mathbf{k} \prec \mathbf{j}} (e_{\mathbf{i},\mathbf{k}} + e_{\mathbf{k},\mathbf{j}}) \geq \sum_{\mathbf{k}} \alpha_{\mathbf{k}} \frac{1}{\min\{\mathcal{M}(\mathbf{i},\mathbf{k}), \mathcal{N}(\mathbf{k},\mathbf{j})\}} \mathcal{M}(\mathbf{i},\mathbf{j}) \binom{d}{\mathbf{j}} \\
&\geq \min_{\mathbf{k}} \frac{1}{\min\{\mathcal{M}(\mathbf{i},\mathbf{k}), \mathcal{N}(\mathbf{k},\mathbf{j})\}} \mathcal{M}(\mathbf{i},\mathbf{j}) \binom{d}{\mathbf{j}} \\
&= \min_{\mathbf{k}} \frac{1}{\mathcal{M}(\mathbf{i},\mathbf{k}) \mathcal{N}(\mathbf{k},\mathbf{j})} \mathcal{M}(\mathbf{i},\mathbf{j}) \binom{d}{\mathbf{j}} \max\{\mathcal{M}(\mathbf{i},\mathbf{k}), \mathcal{N}(\mathbf{k},\mathbf{j})\}.
\end{aligned}
$$

Since this holds for arbitrary $\mathbf{i}$ and $\mathbf{j}$, the size of the 2-TC-spanner is $|S| \geq \mathcal{B}(m,d)$. □

References

[1] N. Ailon, B. Chazelle, S. Comandur, and D. Liu. Property-preserving data reconstruction. *Algorithmica*, 51(2):160–182, 2008.

[2] N. Alon and B. Schieber. Optimal preprocessing for answering on-line product queries. Technical Report 71/87, Tel-Aviv University, 1987.

[3] M. J. Atallah, K. B. Frikken, N. Fazio, and M. Blanton. Dynamic and efficient key management for access hierarchies. In *ACM Conference on Computer and Communications Security*, pages 190–202, 2005.

[4] S. Baswana and S. Sen. Approximate distance oracles for unweighted graphs in expected $\tilde{O}(n^2)$ time. *ACM Transactions on Algorithms*, 2(4):557–577, 2006.

[5] A. Bhattacharyya, E. Grigorescu, M. Jha, K. Jung, S. Raskhodnikova, and D. P. Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. In *RANDOM*, pages 448–461, 2010.

[6] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff. Transitive-closure spanners. In *SODA*, pages 932–941, 2009.

[7] B. Chazelle. Computing on a free tree via complexity-preserving mappings. *Algorithmica*, 2:337–361, 1987.

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

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

[10] L. Cowen. Compact routing with minimum stretch. *J. Algorithms*, 38(1):170–183, 2001.

[11] L. Cowen and C. G. Wagner. Compact roundtrip routing in directed networks. *J. Algorithms*, 50(1):79–95, 2004.

[12] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM*, pages 97–108, 1999.

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

[14] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

[15] W. Hesse. Directed graphs requiring large numbers of shortcuts. In *SODA*, pages 665–669, 2003.

[16] D. Peleg and A. A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

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

[18] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *JACM*, 36(3):510–530, 1989.

[19] L. Roditty, M. Thorup, and U. Zwick. Roundtrip spanners and roundtrip routing in directed graphs. In *SODA*, pages 844–851, 2002.

[20] M. Saks and C. Seshadhri. Local monotonicity reconstruction. *SIAM Journal on Computing*, 39(7):2897–2926, 2010.

[21] R. Seidel. Understanding the inverse Ackermann function. Available at `http://cgi.di.uoa.gr/~ewcg06/invited/Seidel.pdf`, 2006.

[22] M. Thorup. On shortcutting digraphs. In *WG*, pages 205–211, 1992.

[23] M. Thorup. Shortcutting planar digraphs. *Combinatorics, Probability & Computing*, 4:287–315, 1995.

[24] M. Thorup. Parallel shortcutting of rooted trees. *J. Algorithms*, 23(1):139–159, 1997.

[25] M. Thorup and U. Zwick. Compact routing schemes. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 1–10, 2001.

[26] M. Thorup and U. Zwick. Approximate distance oracles. *JACM*, 52(1):1–24, 2005.

[27] A. C.-C. Yao. Space-time tradeoff for answering range queries (extended abstract). In *STOC*, pages 128–136, 1982.