SIAM J. DISCRETE MATH.
Vol. 28, No. 1, pp. 127–141

PRESERVING TERMINAL DISTANCES USING MINORS*

ROBERT KRAUTHGAMER†, HUY L. NGUYỄN‡, AND TAMAR ZONDINER†

Abstract. We introduce the following notion of compressing an undirected graph G with (nonnegative) edge-lengths and terminal vertices $R \subseteq V(G)$. A *distance-preserving minor* is a minor G' (of G) with possibly different edge-lengths, such that $R \subseteq V(G')$ and the shortest-path distance between every pair of terminals is exactly the same in G and in G'. We ask: what is the smallest $f^*(k)$ such that every graph G with $k = |R|$ terminals admits a distance-preserving minor G' with at most $f^*(k)$ vertices? Simple analysis shows that $f^*(k) \leq O(k^4)$. Our main result proves that $f^*(k) \geq \Omega(k^2)$, significantly improving on the trivial $f^*(k) \geq k$. Our lower bound holds even for planar graphs G, in contrast to graphs G of constant treewidth, for which we prove that $O(k)$ vertices suffice.

Key words. distance-preserving minor, graph compression, vertex-sparsification, metric embedding

AMS subject classifications. 05C12, 05C83

DOI. 10.1137/120888843

1. Introduction.

A *graph compression* of a graph G is a small graph G^* that preserves certain features (quantities) of G, such as distances or cut values. This basic concept was introduced by Feder and Motwani [FM95], although their definition was slightly different technically. (They require that G^* has fewer edges than G, and that each graph can be quickly computed from the other one.) Our paper is concerned with preserving the selected features of G *exactly* (i.e., lossless compression), but in general we may also allow the features to be preserved approximately.

The algorithmic utility of graph compression is readily apparent—the compressed graph G^* may be computed as a preprocessing step, and then further processing is performed on it (instead of on G) with lower runtime or memory requirement. This approach is clearly beneficial when the compression can be computed very efficiently, say in linear time, in which case it may be performed on the fly, but it is useful also when some computations are to be performed (repeatedly) on a machine with limited resources such as a smartphone, while the preprocessing can be executed in advance on much more powerful machines.

For many features, graph compression was already studied and many results are known. For instance, a *k-spanner* of G is a subgraph G^* in which all pairwise distances approximate those in G within a factor of k [PS89]. Another example, closer in spirit to our own, is a *sourcewise distance preserver* of G with respect to a set of vertices $R \subseteq V(G)$; this is a subgraph G^* of G that preserves (exactly) the distances in G for all pairs of vertices in R [CE06]. We defer the discussion of further examples and

*Received by the editors August 21, 2012; accepted for publication (in revised form) October 14, 2013; published electronically January 28, 2014. A preliminary version of this paper appeared in Proceedings of ICALP 2012 [KZ12]. The current version features a new proof of the main lower bound result (section 3).

http://www.siam.org/journals/sidma/28-1/88884.html

†Computer Science and Applied Mathematics, Weizmann Institute of Science, 763200 Rehovot, Israel (robert.krauthgamer@weizmann.ac.il, tamar.zondiner@gmail.com). The research of these authors was supported in part by The Israel Science Foundation (grant 452/08), by a US-Israel BSF grant 2010418, and by the Citi Foundation.

‡Princeton University, Princeton, NJ 08540 (hlnguyen@princeton.edu). This author's research was supported in part by NSF CCF 0832797.

related notions to section 1.2, and here point out only two phenomena: First, it is common to require G^* to be structurally similar to G (e.g., a spanner is a subgraph of G), and second, sometimes only the features of a subset R need to be preserved (e.g., distances between vertices of R).

We consider the problem of compressing a graph so as to maintain the shortest-path distances among a set R of required vertices. From now on, the required vertices will be called *terminals*.

DEFINITION 1.1. *Let G be a graph with edge lengths $\ell : E(G) \to \mathbb{R}_+$ and a set of terminals $R \subseteq V(G)$. A* distance-preserving minor *(of G with respect to R) is a graph G' with edge lengths $\ell' : E(G') \to \mathbb{R}_+$ satisfying:*

1. *G' is a minor of G (which means that a graph isomporphic to G' can be obtained from G by a sequence of edge contractions, edge deletions, and vertex deletions); and*
2. *$d_{G'}(u, v) = d_G(u, v)$ for all $u, v \in R$.*

Here and throughout, d_H denotes the shortest-path distance in a graph H. It also goes without saying that the terminals R must survive the minor operations (they are not removed, but might be merged with nonterminals, due to edge contractions), and thus $d_{G'}(u, v)$ is well-defined; in particular, $R \subseteq V(G')$. For illustration, suppose G is a path of n unit-length edges and the terminals are the path's endpoints; then by contracting all the edges, we can obtain G' that is a single edge of length n.

The above definition basically asks for a minor G' that preserves all terminal distances exactly. The minor requirement is a common method to induce structural similarity between G' and G, and in general excludes the trivial solution of a complete graph on the vertex set R (with appropriate edge lengths). The above definition may be viewed as a conceptual contribution of our paper, and indeed our main motivation is its mathematical elegance, but for completeness we also present potential algorithmic applications in section 1.3.

We raise the following question, which to the best of our knowledge was not studied before. Its main point is to bound the size of G' independently of the size of G.

QUESTION 1.2. *What is the smallest $f^*(k)$, such that for every graph G with k terminals, there is a distance-preserving minor G' with at most $f^*(k)$ vertices?*

Before describing our results, let us provide a few initial observations, which may well be folklore or appear implicitly in literature. There is a *naive algorithm* which constructs G' from G by two simple steps (Algorithm 1 in section 2):

(1) Remove all vertices and edges in G that do not participate in any shortest-path between terminals.
(2) Repeat while the graph contains a nonterminal v of degree two: merge v with one of its neighbors (by contracting the appropriate edge), thereby replacing the 2-path $w_1 - v - w_2$ with a single edge (w_1, w_2) of the same length as the 2-path.

It is straightforward to see that these steps reduce the number of nonterminals without affecting terminal distances, and a simple analysis proves that this algorithm always produces a minor with $O(k^4)$ vertices and edges (and runs in polynomial time). It follows that $f^*(k)$ exists, and moreover

$$f^*(k) \le O(k^4).$$

Furthermore, if G is a tree, then G' has at most $2k-2$ vertices, and this last bound is in fact tight (obtained by a complete binary tree) whenever k is a power of 2. We are not aware of explicit references for these analyses, and thus review them in section 2.

1.1. Our results. Our first and main result directly addresses Question 1.2, by providing the lower bound $f^*(k) \geq \Omega(k^2)$. The proof uses a simple planar graph (a two-dimensional grid), leading us to study the restriction of $f^*(k)$ to specific graph families, defined as follows.[1]

DEFINITION 1.3. *For a family $\mathcal{F}$ of graphs, define $f^*(k, \mathcal{F})$ as the minimum integer such that every graph $G = (V, E, \ell) \in \mathcal{F}$ with k terminals admits a distance-preserving minor G' with at most $f^*(k, \mathcal{F})$ vertices.*

THEOREM 1.4. *Let* Planar *be the family of all planar graphs. Then*

$$f^*(k) \geq f^*(k, \mathsf{Planar}) \geq \Omega(k^2).$$

Moreover, our proof shows that this lower bound extends to the case where distances are preserved within some approximation (rather than exactly). Specifically, we prove in section 3 that for every $\varepsilon \in [\frac{1}{4\lfloor k/4 \rfloor}, \frac{1}{4}]$, there exists a planar graph G with at most k terminals such that every minor G' that approximates the terminal distances within a $(1+\varepsilon)$ factor must have $\Omega(\varepsilon^{-2})$ nonterminals. We remark that aour original proof of this theorem [KZ12, Theorem 3] required all terminal distances to be preserved exactly (rather than approximately), although it only required G' to be planar (rather than a minor of G).

Our proof of Theorem 1.4 uses $k \times k$ grid graphs, whose treewidth is k, and thus not bounded by a constant. (The definition of treewidth, along with basic properties, can be found in standard texts [Bodlaender06, Kloks94]; our results do not use this definition directly.) This stands in contrast to graphs of treewidth 1, because we already mentioned that

$$f^*(k, \mathsf{Trees}) \leq 2k - 2,$$

where Trees is the family of all tree graphs. It is thus natural to ask whether bounded-treewidth graphs behave like trees, for which $f^* \leq O(k)$, or like planar graphs, for which $f^* \geq \Omega(k^2)$. We answer this question as follows.

THEOREM 1.5. *Let* Treewidth(p) *be the family of all graphs with treewidth at most p. Then for all $k \geq p$,*

$$\Omega(pk) \leq f^*(k, \mathsf{Treewidth}(p)) \leq O(p^3 k).$$

We summarize our results together with some initial observations in the table below.

Graph family $\mathcal{F}$	Bounds on $f^*(k, \mathcal{F})$		Reference
Trees	$= 2k-2$		Theorems 2.4, 2.3
Treewidth p	$\Omega(pk)$	$O(p^3k)$	Theorem 1.5
Planar graphs	$\Omega(k^2)$	$O(k^4)$	Theorems 1.4, 2.1
All graphs	$\Omega(k^2)$	$O(k^4)$	Theorems 1.4, 2.1

All our upper bounds are algorithmic and run in polynomial time. In fact, they can be achieved using the naive algorithm (Algorithm 1 in section 2).

1.2. Related work. Coppersmith and Elkin [CE06] studied a problem similar to ours, except that they seek subgraphs with few edges (rather than minors). Among other things, they prove that for every weighted graph $G = (V, E)$ and every set of $k =$

[1] We use (V, E, ℓ) to denote a graph with vertex set V, edge set E, and edge lengths $\ell : E \to \mathbb{R}_+$. As usual, the definition of a family $\mathcal{F}$ of graphs refers only to the vertices and edges, and is irrespective of the edge lengths.

$O(|V|^{1/4})$ terminals (sources), there exists a weighted subgraph $G' = (V, E')$, called a *sourcewise preserver*, that preserves terminal distances exactly and has $|E'| \leq O(|V|)$ edges. They also show a nearly matching lower bound on $|E'|$. The question of preserving the distances between terminals in G using a small graph G' that is not required to be "similar to" (e.g., a minor or subgraph of) G, was studied under the terminology of Steiner spanners or emulators [ADDJS93, RTZ05, Woodruff06].

Some compression methods preserve cuts and flows in a given graph G rather than distances. A Gomory–Hu tree [GH61] is a weighted tree that preserves all st-cuts in G (or just between terminal pairs). A so-called mimicking network preserves all flows and cuts between subsets of the terminals in G [HKNR98].

Terminal distances can also be approximated instead of preserved exactly. In fact, allowing a constant factor approximation may be sufficient to obtain a compression G^* without any nonterminals. Gupta [Gupta01] introduced this problem and proved that for every weighted tree T and set of terminals, there exists a weighted tree T' without the nonterminals that approximates all terminal distances within a factor of 8. It was later observed that this T' is in fact a minor of T [CGNRS06], and that the factor 8 is tight [CXKR06]. Basu and Gupta [BG08] claimed that a constant approximation factor exists for weighted outerplanar graphs as well. It remains an open problem whether the constant factor approximation extends also to planar graphs (or excluded-minor graphs in general). Englert et al. [EGKRTT10] proved a randomized version of this problem for all excluded-minor graph families, with an expected approximation factor depending only on the size of the excluded minor.

The relevant information (features) in a graph can also be maintained by a data structure that is not necessarily graphs. A notable example is distance oracles—low-space data structures that can answer distance queries (often approximately) in constant time [TZ05]. These structures adhere to our main requirement of "compression" and are designed to answer queries very quickly. However, they might lose properties that are natural in graphs, such as the triangle inequality or the similarity of a minor to the given graph, which may be useful for further processing of the graph.

1.3. Potential applications. Our first example application is in the context of algorithms dealing with graph distances. Often, algorithms that are applicable to an input graph G are applicable also to a minor of it G' (e.g., algorithms for planar graphs). Consider for instance the traveling salesman problem, which is known to admit a quasi-polynomial time approximation scheme (QPTAS) in excluded-minor graphs [GS02] (and polynomial time approximation scheme (PTAS) in planar graphs [Klein08]), even if the input contains a set of *clients* (a subset of the vertices that must be visited by the tour). Suppose now that the clients change daily, but they can only come from a fixed and relatively small set $R \subset V(G)$ of potential clients. Obviously, once a distance-preserving minor G' of G is computed, the QPTAS can be applied on a daily basis to the small graph G' (instead of to G). Notice how important it is to preserve all terminal distances exactly using G' that is a minor of G. (A complete graph on vertex set R would not work, because we do not have a QPTAS for it.)

Our second example application is in the field of metric embeddings. Consider a known embedding, such as the embedding of a bounded-genus graph G into a distribution over planar graphs [IS07]. Suppose we want to use this embedding, but we only care about a small subset of the vertices $R \subset V(G)$. We can compute a distance-preserving minor G' (and thus with same genus bound) that has at most $f^*(|R|)$ vertices, and then apply the known embedding to the small graph G' (instead of to G). The result would be a distribution over planar graphs, each of them having at most $f^*(|R|)$ vertices, independently of $|V(G)|$. In (other) cases where the embedding's distortion depends on $|V(G)|$, this approach may even yield improved distortion bounds, such as replacing $O(\log|V(G)|)$ terms with $O(\log|R|)$.

2. Review of straightforward analyses. As described in the introduction, a naive way to create a minor G' of G preserving terminal distances is to perform the steps described in ReduceGraphNaive, depicted below as Algorithm 1. In this section, we show that for general graphs G, the returned minor has at most $O(k^4)$ vertices, and for trees it has at most $2k-2$ vertices. We assume henceforth that ties between shortest paths (connecting the same pair of vertices) are broken in a *consistent* way, in the sense that whenever Π is the "chosen" shortest path (between its endpoints), every subpath of Π is also the "chosen" shortest path. Such tie breaking can be implemented, for example, by applying small perturbations to the edge-weights to make sure that no two paths in the graph have the exact same length.

Algorithm 1. ReduceGraphNaive (graph G, required vertices R).

1: Compute shortest paths between every pair of terminals, breaking ties consistently.
2: Remove nonterminals and edges that do not participate in any terminal-to-terminal shortest path.
3: **while** there exists a nonterminal v incident to only two edges (v,u) and (v,w) **do**
4: contract the edge (u,v),
5: set the length of edge (u,w) to be $d_G(u,w)$.

It is easy to see that G' is a distance-preserving minor of G with respect to R. The time complexity of this algorithm is at most that of k applications of single-source shortest paths, plus $O(|E|+kn)$ time for reading the resulting k trees, removing unnecessary edges and vertices and contracting edges connecting vertices with degree 2 with their neighbors in the shortest path.

2.1. $f^*(k) \le O(k^4)$ for general graphs.

Theorem 2.1. *For every graph G and set $R \subseteq V$ of k terminals, the output G' of* ReduceGraphNaive(G,R) *is a distance-preserving minor of G with at most $O(k^4)$ vertices. In particular, $f^*(k) \le O(k^4)$.*

To prove the theorem, we will need the following lemma, whose proof is sketched below. A more detailed proof is given in [CE06, Lemma 7.5], where the lemma is used to bound the number of edges in the graph G' after only performing on a graph the edge-removals in line 1 of ReduceGraphNaive.

Lemma 2.2. *Let G be a graph, and suppose that ties between shortest paths (connecting the same pair of vertices) are broken in a consistent way. Then every two distinct shortest paths between terminals in G, denoted Π and Π', branch in at most two vertices, i.e., there at most two vertices $v \in V(\Pi)\cap V(\Pi')$ such that $|N_{\Pi\cup\Pi'}(v)|$ (the number of neighbors of v in both paths) is greater than 2.*

Proof of Sketch. Using the path Π in some direction to determine the order, denote by v_1 and v_2 the first and last vertices on Π such that $|N_{\Pi\cup\Pi'}(v_1)|, |N_{\Pi\cup\Pi'}(v_2)| > 2$. We call such vertices "branching vertices." Since Π and Π' are shortest paths, in either of them v_1 and v_2 have at most two neighbors, and hence these vertices appear in both paths. In fact, both Π and Π' have a subpath connecting v_1 and v_2. Since this subpath in both Π and Π' is the shortest path between v_1 and v_2 in G, and we assumed that the shortest path in G is unique, it must hold that the subpath $p(v_1,v_2)$ is shared by both Π and Π', implying that any other vertex u on that subpath has

$|N_{\Pi \cup \Pi'}(u)| = 2$, and v_1, v_2 are the only such vertices on the subpath. Since they are the first and last vertices on Π, they are in fact the only ones in the whole path, thus proving the lemma. ∎

Proof of Theorem 2.1. Every nonterminal $v \in V' \setminus R$ has degree greater or equal to 3, and hence it is a branching vertex. Every pair of shortest paths contributes at most two branching vertices to G'. There are $O(k^4)$ such pairs, and therefore $O(k^4)$ vertices in V'. Since G' is also a distance-preserving minor of G with respect to R, this completes the proof of Theorem 2.1. ∎

It is interesting to note that G' is relatively sparse; the number of edges in the graph is at most $O(k^4)$, which matches our bound on the number of vertices. Indeed, the edges of G' all lie on shortest paths between terminals, and since there are $\binom{k}{2}$ such paths, it suffices to prove that every shortest path between two terminals contains at most $O(k^2)$ edges. To this end, fix a shortest path $\Pi(t_1, t_2)$ between two terminals t_1, t_2. Consider a nonterminal vertex v on this path, which by construction must be a branching vertex for two or more shortest paths. Assume toward contradiction that v is a branching vertex for two *other* shortest paths Π_1 and Π_2, but not a branching vertex for $\Pi(t_1, t_2)$ with either of Π_1 or Π_2. The path $\Pi(t_1, t_2)$ contains v as an internal vertex, and hence $|N_{\Pi(t_1,t_2)}(v)| = 2$. Since v is not a branching vertex for $\Pi(t_1, t_2)$ and Π_1, those two edges are shared by both paths, i.e., $|N_{\Pi(t_1,t_2)\cup\Pi_1}(v)| = 2$. The same, of course, applies to Π_2, and thus $|N_{\Pi_1\cup\Pi_2}(v)| = 2$, which contradicts v being a branching vertex for Π_1 and Π_2. It follows that every nonterminal vertex v on $\Pi(t_1, t_2)$ must be a branching vertex for $\Pi(t_1, t_2)$ and another shortest path between two terminals. Since there are at most two branching vertices for any two shortest paths, the total number of nonterminal vertices on $\Pi(t_1, t_2)$ is at most $2(\binom{k}{2} - 1)$. We conclude that each path $\Pi(t_1, t_2)$ contains at most $O(k^2)$ edges, and this proves that the total number of edges in G' is $O(k^4)$.

2.2. $f^*(k, \textsf{Trees}) = 2k - 2$.

THEOREM 2.3. *For every tree G and set $R \subseteq V$ of k terminals, the output G' of* REDUCEGRAPHNAIVE(G, R) *is a distance-preserving minor of G with at most $2k-2$ vertices. In particular, $f^*(k, \textsf{Trees}) \leq 2k - 2$.*

Proof. Every nonterminal $v \in V' \setminus R$ has degree greater or equal to 3. Let s denote the number of nonterminals in the tree G'. Then

$$\sum_{v \in V'} \deg_{G'}(v) \geq k + 3s.$$

Since G' is a tree, the sum of its degrees also equals $2(k+s)-2$, and hence $2(k+s)-2 \geq k + 3s$, and $s \leq k - 2$, proving the theorem. ∎

This bound is exactly tight. We sketch the proof of the following theorem.

THEOREM 2.4. *For every $i \in \mathbb{N}$ there exists a tree G and $k = 2^i$ terminals $R \subseteq V$ such that every distance-preserving minor G' of G with respect to R has $|V'| \geq 2k-2$. In particular, $f^*(k, \textsf{Trees}) \geq 2k - 2$ for $k = 2^i$.*

Proof. Consider the complete binary tree G of depth i with unit edge-lengths. Let the 2^i leaves of the tree be the terminals R. We use induction on i to prove that for the complete binary tree with level i, the only edge contraction (and indeed the only minor operation) allowed is the contraction of an edge between the root and one of its children.

In the tree with depth 1, this is clearly true. Let T be the complete binary tree with depth $i + 1$, and T_1, T_2 be its two i-depth subtrees. Assume toward contraction that the distance preserving minor T' was created by contraction operations which ended up merging a vertex v_1 in T_1 with a vertex v_2 in T_2. Denote the merged vertex as x. From the binary tree structure, x is the only vertex that connects vertices in T_1 with vertices in T_2. It is easy to see that v_1 participates in some shortest path between leaves (terminals) $t_1, t_2 \in T_1$ and v_2 participates in some shortest path between terminals $s_1, s_2 \in T_2$. Let t_1 and s_1 be the vertices within a smaller distance from x. Then

$$\begin{aligned} d_{T'}(t_1, s_1) &= d_{T'}(t_1, x) + d_{T'}(x, s_1) \\ &\le \frac{1}{2} d_{T'}(t_1, t_2) + \frac{1}{2} d_{T'}(s_1, s_2) \\ &= \frac{1}{2} d_{T_1}(t_1, t_2) + \frac{1}{2} d_{T'}(s_1, s_2) \le \max\{d_{T_1}(t_1, t_2), d_{T_2}(s_1, s_2)\}. \end{aligned}$$

However, $d_{T'}(t_1, s_1) = d_T(t_1, s_1) = 2(i+1)$, whereas $\max\{d_{T_1}(t_1, t_2), d_{T_2}(s_1, s_2)\} \le 2i$, which contradicts any contractions combining vertices from T_1 and T_2. Within (without loss of generality) T_1, the induction hypothesis holds; therefore, the only contraction to be considered is that of the root of T_1 with one of its children. However, such a contraction is not possible while retaining the equal distance $2(i+1)$ between every terminal $t \in T_1$ and $s \in T_2$. Therefore, no contractions can occur inside T_1 and T_2. The root of T can be joined with either of its children (the roots of T_1 and T_2), but not both (since they cannot be joined as a single vertex), thus proving the theorem. ☐

3. A lower bound of $\Omega(k^2)$.

In this section we prove Theorem 1.4, stated with more details as follows.

THEOREM 3.1. *For every integer $k \ge 4$, there is a planar graph G with k terminals, such that every distance-preserving minor of G has $\Omega(k^2)$ nonterminals. In fact, this graph G is just an unweighted $O(k) \times O(k)$ grid, with all its terminals on the boundary.*

We actually prove a slightly stronger theorem, allowing distances to be distorted by a $1+\varepsilon$ factor, and therefore need the following variant of Definition 1.1.

DEFINITION 3.2. *Let G be a graph with edge lengths $\ell : E(G) \to \mathbb{R}_+$ and a set of terminals $R \subseteq V(G)$, and let $\alpha \ge 1$. An* α-distance-approximating minor *(of G with respect to R) is a graph G' with edge lengths $\ell' : E(G') \to \mathbb{R}_+$ satisfying*

1. *G' is a minor of G; and*
2. *$d_G(u,v) \le d_{G'}(u,v) \le \alpha \cdot d_G(u,v)$ for all $u, v \in R$.*

It is easy to check that Theorem 3.1 is a corollary of the following theorem for $\varepsilon = \frac{1}{4\lfloor k/4 \rfloor}$.

THEOREM 3.3. *For every $\varepsilon > 0$ for which $1/(4\varepsilon)$ is an integer, there is a planar graph G with $1/\varepsilon$ terminals, such that every $(1+\varepsilon)$ distance-approximating minor of G must have $\Omega(1/\varepsilon^2)$ nonterminals. In fact, this graph G is just an unweighted $O(1/\varepsilon) \times O(1/\varepsilon)$ grid, with all its terminals on the boundary.*

Proof. Let $1/\varepsilon = 4r$ for some integer $r \ge 1$. Let G be a two-dimensional grid of size $(r+2) \times (r+2)$, and let all the boundary vertices not at the corners be its terminals. Clearly, there are $r = 1/(4\varepsilon)$ terminals on each side of the grid.

Consider an edge-weighted graph G' that is a $(1+\varepsilon)$ distance-approximating minor of G. Let $x_1, \ldots, x_r$ denote the terminals on the left side of the grid G ordered from top to bottom, and similarly $y_1, \ldots, y_r$ for the right side. Recall that G' also contains all these terminals, and let P'_i be a shortest path in G' between x_i and y_i. We shall refer to these r paths as "horizontal."

Claim 3.4. *The horizontal shortest paths $P'_1, \ldots, P'_r$ do not intersect each other (i.e., they are vertex-disjoint).*

Proof. Fix $i < j$, and assume to the contrary that the paths P'_i, P'_j intersect at some vertex $z \in V(G')$. Then by the $1+\varepsilon$ distance approximation guarantee,

$$d_{G'}(x_i, z) + d_{G'}(z, y_i) = d_{G'}(x_i, y_i) \le (1+\varepsilon)(r+1) < r+2, \tag{3.1}$$

$$d_{G'}(x_j, z) + d_{G'}(z, y_j) = d_{G'}(x_j, y_j) \le (1+\varepsilon)(r+1) < r+2. \tag{3.2}$$

Now consider the "cross distances" from x_i to y_j, and similarly from x_j to y_i (again in G'). We can bound the two cross distances using the triangle inequality and the intersection vertex z,

$$d_{G'}(x_i, y_j) \le d_{G'}(x_i, z) + d_{G'}(z, y_j),$$
$$d_{G'}(x_j, y_i) \le d_{G'}(x_j, z) + d_{G'}(z, y_i).$$

Summing these two inequalities and plugging in (3.1) and (3.2), we obtain

$$d_{G'}(x_i, y_j) + d_{G'}(x_j, y_i) < 2(r+2).$$

However, we can compute the corresponding distances in G directly

$$d_G(x_i, y_j) = d_G(x_j, y_i) = r + 1 + |i-j| \ge r+2.$$

The last two inequalities contradict the assumption that $d_{G'}$ dominates d_G, and the claim follows. ∎

The above argument applies also to the "vertical" paths $Q'_1, \ldots, Q'_r$. Formally, let Q'_i be a shortest path in G' between the ith terminal on the top boundary and the ith terminal on the bottom boundary (ordered from left to right).

Claim 3.5. *The vertical shortest paths $Q'_1, \ldots, Q'_r$ do not intersect each other (i.e., they are vertex-disjoint).*

We next claim that all the horizontal paths must intersect all the vertical paths.

Claim 3.6. *Every horizontal path P'_i and every vertical path Q'_j intersect (in G').*

Proof. Because G' is a minor of G, a path in G' that connects two terminals can be mapped back to a path in G connecting the same two terminals. Thus, the path P'_i is mapped to a path in G, which we denote P_i, and similarly Q'_j is mapped back to a path in G denoted Q_j. It is easy to verify that P_i and Q_j must intersect at some vetex of G, and forward mapping this vertex into the minor G' gives an intersection vertex between P'_i and Q'_j. ∎

We can now complete the proof of Theorem 3.3. Paths P'_i and Q'_j must intersect by Claim 3.6, so let us denote the intersection vertex by $z'_{i,j} \in V(G')$ (choosing one arbitrarily if there are multiple ones). We argue that all these vertices must be distinct. Indeed, suppose $z_{i_1,j_1} = z_{i_2,j_2}$. Then this vertex belongs to both P'_{i_1} and P'_{i_2}, which, using Claim 3.4, implies that $i_1 = i_2$. A similar argument implies that $j_1 = j_2$. We conclude that $z_{i,j} \in V(G)$ for $i, j \in [r]$ are all distinct vertices. Moreover, P'_i and Q'_j cannot intersect at a terminal, because it implies that $d_{G'}(x_i, y_i) \ge d_G(x_i, y_i) + 2 = (1 + \frac{2}{r+1}) d_G(x_i, y_i)$, leading to violation of the $1+\varepsilon$ approximation. This proves that G contains at least $r^2 = 1/(16\varepsilon^2)$ nonterminals. ∎

A $k \times k$ grid graph has treewidth (exactly) k, and we thus immediately obtain the following corollary, which we record here for later use in section 4.2.

Corollary 3.7. *For every $p \in \mathbb{N}$ there exists a graph G with treewidth p and p terminals $R \subseteq V$, such that every distance-preserving minor G' of G with respect to R has $|V(G')| \ge \Omega(p^2)$.*

4. $\Theta(k)$ bounds for constant treewidth graphs. In this section we prove Theorem 1.5, which bounds $f^*(k, \mathsf{Treewidth}(p))$. The upper and the lower bound are proved separately in Theorems 4.1 and 4.7 below.

4.1. An upper bound of $O(p^3 k)$.

THEOREM 4.1. *Every graph $G = (V, E, \ell)$ with treewidth p and a set $R \subseteq V$ of k terminals admits a distance-preserving minor $G' = (V', E', \ell')$ with $|V'| \leq O(p^3 k)$. In other words, $f^*(k, \mathsf{Treewidth}(p)) \leq O(p^3 k)$.*

The graph G' can in fact be computed in time polynomial in $|V|$ (see Corollary 4.6).

Without loss of generality, we may assume that $k \geq p$, since otherwise the $O(k^4)$ bound from Theorem 2.1 applies. To prove Theorem 4.1, we introduce the algorithm REDUCEGRAPHTW (depicted in Algorithm 2 below), which follows a divide-and-conquer approach. We use the small separators guaranteed by the treewidth p, to break the graph recursively until we have small, almost-disjoint subgraphs. We apply the naive algorithm (REDUCEGRAPHNAIVE, depicted in Algorithm 1 in section 2) on each of these subgraphs with an altered set of terminals—the original terminals in the subgraph, plus the separator (*boundary*) vertices which disconnect these terminals from the rest of the graph. We get many small distance-preserving minors, which are then combined into a distance-preserving minor G' of the original graph G.

Proof of Theorem 4.1. The divide-and-conquer technique works as follows. Given a partitioning of V into the sets A_1, S, and A_2, such that removing S disconnects A_1 from A_2, the graph G is divided into the two subgraphs $G[A_i \cup S]$ (the subgraph of G induced on $A_i \cup S$) for $i \in \{1, 2\}$. For each $G[A_i \cup S]$, we compute a distance-preserving minor with respect to terminals set $(R \cap A_i) \cup S$, and denote it $\hat{G}_i = (\hat{V}_i, \hat{E}_i, \hat{\ell}_i)$. The two minors are then combined into a distance-preserving minor of G with respect to R, according to the following definition.

We define the *union* $H_1 \cup H_2$ of two (not necessarily disjoint) graphs $H_1 = (V_1, E_1, \ell_1)$ and $H_2 = (V_2, E_2, \ell_2)$ to be the graph $H = (V_1 \cup V_2, E_1 \cup E_2, \ell)$, where the edge lengths are $\ell(e) = \min\{\ell_1(e), \ell_2(e)\}$ (assuming infinite length when $\ell_i(e)$ is undefined). A crucial point here is that H_1, H_2 need not be disjoint—overlapping vertices are merged into one vertex in H, and overlapping edges are merged into a single edge in H.

LEMMA 4.2. *The graph $\hat{G} = \hat{G}_1 \cup \hat{G}_2$ is a distance-preserving minor of G with respect to R.*

Proof of Lemma 4.2. Note that since the *boundary vertices* in S exist in both $\hat{G}_1$ and $\hat{G}_2$, they are never contracted into other vertices. In fact, the only minor-operation allowed on vertices in S is the removal of edges (s_1, s_2) for two vertices $s_1, s_2 \in S$, when shorter paths in $G[A_1 \cup S]$ or $G[A_2 \cup S]$ are found. It is thus possible to perform both sequences of minor-operations independently, making $\hat{G}$ a minor of G.

A path between two vertices $t_1, t_2 \in R$ can be split into subpaths at every visit to a vertex in $R \cup S$, so that each subpath between $v, u \in R \cup S$ does not contain any other vertices in $R \cup S$. Since there are no edges between A_1 and A_2, each of these subpaths exists completely inside $G[A_1 \cup S]$ or inside $G[A_2 \cup S]$. Hence, for every subpath between $v, u \in R \cup S$, it holds that $d_G(v, u) = d_{G[A_i \cup S]}(v, u) = d_{\hat{G}_i}(v, u)$ for some $i \in \{1, 2\}$. Applying this argument to a shortest path between $t_1, t_2 \in R$ in G, we obtain a path between them in $\hat{G}$, and it follows that $d_{\hat{G}}(t_1, t_2) \leq d_G(t_1, t_2)$.

A similar argument can be made in the opposite direction, by considering a shortest path between t_1, t_2 in $\hat{G}$, splitting it as necessary and eventually obtaining a path of the same length in G. Hence, $\hat{G}$ is a distance-preserving minor of G. ∎

A well-known consequence of the graph G having bounded treewidth p, see, e.g., [BGHK95, Lemma 6] for a similar claim,[2] is that for every nonnegative vertex-weights $w(\cdot)$, there exists a set $S \subseteq V$ of at most $p+1$ vertices (to simplify the analysis, we assume this number is p) whose removal separates the graph into two parts, A_1 and A_2, each with $w(A_i) \leq \frac{2}{3}w(V)$. It is then natural to compute a distance-preserving minor for each part A_i by recursion, and then combine the two solutions using Lemma 4.2. We can use the weights $w(\cdot)$ to obtain a balanced split of the terminals, and thus $|R \cap A_i|$ is a constant factor smaller than $|R|$. However, when solving each part A_i, the boundary vertices S must be counted as "additional" terminals, and to prevent those from accumulating too rapidly, we compute (à la [Bodlaender89]) a second separator S^i with different weights $w(\cdot)$ to obtain a balanced split of the boundary vertices accumulated so far.

Algorithm ReduceGraphTW receives, in addition to a graph H and a set of terminals $R \subseteq V(H)$, a set of boundary vertices $B \subseteq V(H)$. Note that a terminal that is also on the boundary is counted only in B and not in R, so that $R \cap B = \emptyset$.

The procedure Separator(H, U) returns the triple $\langle A_1, S, A_2 \rangle$ of a separator S and two sets A_1 and A_2 such that $|S| \leq p$, no edges between A_1 and A_2 exist in G, and $|A_1 \cap U|, |A_2 \cap U| \leq \frac{2}{3}|U|$, i.e., using $w(\cdot)$ that is unit-weight inside U, and 0 otherwise.

The algorithm works as follows. In Line 3 the separator is computed in such a way that the terminals are balanced between the two sets A_1 and A_2. In lines 4–10, distance-preserving minors $\hat{G}_1$ and $\hat{G}_2$ are generated for the subgraphs $H[A_1 \cup S]$ and $H[A_2 \cup S]$. In line 11 the two distance-preserving minors are combined. Looking further into lines 4–10, in line 5 the second separation is performed; in this case the balance is over the boundary vertices. Lines 6–7 establish the required vertices and boundary vertices in the resulting two subgraphs of $H[A_i \cup S]$. In line 9 a recursive call is made to ReduceGraphTW with these subgraphs. In line 10 these subgraphs are combined to form $\hat{G}_i$. The recursion is terminated at the stop condition on lines 1–2, where a distance preserving minor is computed by algorithm ReduceGraphNaive.

Algorithm 2. ReduceGraphTW (graph H, required vertices R, boundary vertices B).

1: **if** $|R \cup B| \leq 18p$ **then**
2:     **return** ReduceGraphNaive$(H, R \cup B)$ (see Algorithm 1)
3: $\langle A_1, S, A_2 \rangle \leftarrow$ Separator(H, R)
4: **for** $i = 1, 2$ **do**
5:     $\langle A_i^1, S^i, A_i^2 \rangle \leftarrow$ Separator$(H[A_i \cup S], (B \cap A_i) \cup S)$
6:     $R^i \leftarrow R \setminus (S \cup S^i)$
7:     $B^i \leftarrow B \cup S \cup S^i$
8:     **for** $j = 1, 2$ **do**
9:         $\hat{G}_i^j \leftarrow$ ReduceGraphTW$(H[A_i^j \cup S^i], R^i \cap A_i^j, B^i \cap (A_i^j \cup S^i))$
10:         $\hat{G}_i \leftarrow \hat{G}_i^1 \cup \hat{G}_i^2$
11: **return** $\hat{G}_1 \cup \hat{G}_2$.

See Figure 1 for an illustration of a single execution. Consider the recursion tree T on this process, starting with the invocation of ReduceGraphTW$(G, R, \emptyset)$. A

[2]The claim there is slightly different. First, it is for $0-1$ weights, but the proof extends immediately to arbitrary nonnegative weights. Second, it proves that every connected component of $G \setminus S$ has weight at most $\frac{1}{2}w(V)$; our assertion above follows by repeatedly aggregating the two parts of least weight.

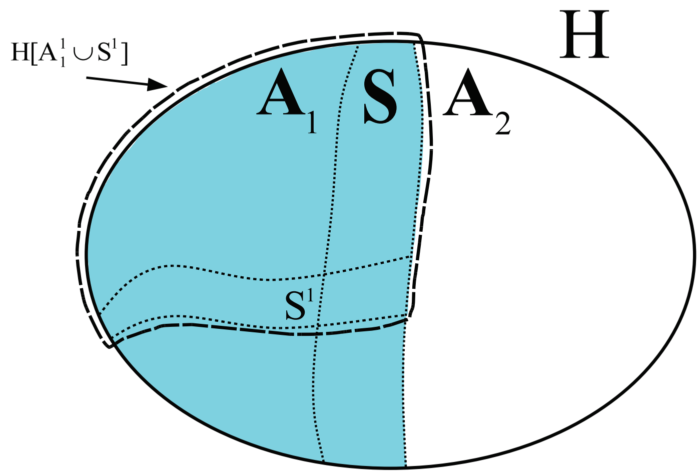

FIG. 1. *The separators S (from line 3) and S^1 (from line 5), and the subgraph $H[A_1^1 \cup S^1]$ to be processed recursively (in line 9).*

node $a \in V(T)$ corresponds to an invocation ReduceGraphTW(H_a, R_a, B_a). The execution either terminates at line 2 (the stop condition) or performs 4 additional invocations b_i for $i \in [1, 4]$, each with $|R_{b_i}| \le \frac{2}{3}|R_a|$. As the process continues, the number of terminals in R_a decreases, whereas the number of boundary vertices may increase. We show the following upper bound on the number of boundary vertices B_a.

LEMMA 4.3. *For every $a \in V(T)$, the number of boundary vertices $|B_a| < 6p$.*

Proof of Lemma 4.3. Proceed by induction on the depth of the node in the recursion tree. The lemma clearly holds for the root of the recursion-tree, since initially $B = \emptyset$. Suppose it holds for an execution with values H_a, R_a, B_a. When partitioning $V(H_a)$ into A_1, S, and A_2, the separator S has at most p vertices. From the induction hypothesis, $|B_a| < 6p$, making $|B_a \cup S| < 7p$.

The algorithm constructs another separator, this time separating the boundary vertices $B_a \cup S$. For $i = 1, 2$ and $j = 1, 2$, it holds that $|S^i| \le p$, $|A_i^j| \le \frac{2}{3} \cdot |B_a \cup S| \le \frac{2}{3} \cdot 7p = \frac{14}{3}p$, and so $|A_i^j \cup S^i| \le \frac{14}{3}p + p < 6p$. The execution corresponding to the node a either terminates in line 2 or invokes executions with the values $A_i^j \cup S^i$ for $i, j = 1, 2$; hence all new invocations have less than $6p$ boundary vertices. ∎

We also prove the following lower bound on the number of terminals R_a.

LEMMA 4.4. *Every $a \in V(T)$ is either a leaf of the tree T or it has at least two children, denoted b_1, b_2, such that $|R_{b_1}|, |R_{b_2}| \ge p$.*

Proof of Lemma 4.4. Consider a node $a \in V(T)$. If this execution terminates at line 2, a is a leaf and the lemma is true. Otherwise it holds that $|R_a \cup B_a| \ge 18p$. Since Lemma 4.3 states that $|B_a| \le 6p$, it must hold that $|R_a| \ge 12p$.

When performing the separation of $V(H_a)$ into A_1, S, and A_2, the vertices R_a are distributed between A_1, S, and A_2, such that $|R_a \cap (A_i \cup S)| \ge \frac{1}{3}|R_a| = 4p$ for $i = 1, 2$. Since $|S| \le p$, it must hold that $|(R_a \setminus S) \cap A_i| = |(R_a \cap (A_i \cup S)) \setminus S| \ge 3p$. When the next separation is performed, at most p of these $3p$ terminals belong to S^i, while the remaining terminals belong to R^i and are distributed between A_i^1 and A_i^2. At least one of these sets, without loss of generality A_i^1, gets $|R^i \cap A_i^1| \ge \frac{1}{2}2p = p$. This is a value of R_b for a child b of a in the recursion tree. Since this holds for both A_1 and A_2, at least two invocations b_1, b_2 with $|R_{b_i}| \ge p$ are made. ∎

The following observation is immediate from Lemma 4.3.

OBSERVATION 4.5. *Every node $a \in V(T)$ such that $|R_a| < p$ has $|R_a \cup B_a| \leq 7p$, and thus it is a leaf in T.*

To bound the size of the overall combined graph G' returned by the first call to REDUCEGRAPHTW, we must bound the number of leaves in T. To do that, we first consider the recursion tree T' created by removing those nodes a with $|R_a| < p$; these are leaves from Observation 4.5. From Lemma 4.4, every node in this tree (except the root) is either a leaf (with degree 1) or has at least two children (with degree at least 3). Since the average degree in a tree is less than 2, the number of nodes with degree at least 3 is bounded by the number of leaves. Every leaf b in the tree T' has $|R_b| \geq p$. These terminals do not belong to any boundary, so for every other leaf b' in T' it holds that $R_b \cap (R_{b'} \cup B_{b'}) = \emptyset$ and these p terminals are unique. There are k terminals in G, so there are $O(k/p)$ such leaves, and $O(k/p)$ internal nodes.

From Lemma 4.4, invocations are performed only by internal vertices in T'. Each internal vertex has four children; hence there are $O(k/p)$ invocations overall. Each leaf in T has $|R_a \cup B_a| \leq O(p)$; hence the graph returned from REDUCEGRAPHNAIVE(H_a) is a distance-preserving minor with $O(p^4)$ vertices (see section 2). Using Lemma 4.2, the combination of these graphs is a distance-preserving minor $\hat{G}$ of G with respect to R. The minor $\hat{G}$ has $O(k/p \cdot p^4) = O(k \cdot p^3)$ vertices, proving Theorem 4.1. ∎

COROLLARY 4.6. *Given a graph with treewidth p, the naive algorithm returns a distance-preserving minor of size $O(k \cdot p^3)$.*

Proof. If $p > k$, this statement holds by the $O(k^4)$ bound. Otherwise, consider the algortihm REDUCEGRAPHTW. All the operations performed on the graph by this algorithm—edge or vertex removals, and edge contractions—are performed during a call to REDUCEGRAPHNAIVE on subgraphs independent of the entire graph structure, with added constraints regarding the boundary vertices.

If a vertex or an edge is removed, it exists entirely in one component of the separator and does not participate in any shortest path in that component between terminals or boundary vertices. No shortest path between two terminals in the original graph could contain this edge, or in particular it would contain a (shortest) subpath between terminals or boundary vertices, completely contained in the component, that contains this edge. Hence this vertex or edge would have been removed also in a regular execution of the naive algorithm given the terminals. A similar argument also applies for edge contractions—if a vertex has degree 2 inside its component when considering all shortest paths between the terminals and boundary vertices, then no other edge adjacent to it can participate in any shortest path between terminals in the original graph. Hence the edge contraction, or perhaps even deletion of this vertex, would have occurred in a regular execution of the naive algorithm on the original graph with respect to the terminals.

Hence, the naive algorithm obtains a similar or smaller graph than algorithm REDUCEGRAPHTW. ∎

4.2. A lower bound of $\Omega(pk)$.

THEOREM 4.7. *For every p and $k \geq p$, there is a graph $G = (V, E, \ell)$ with treewidth p and k terminals $R \subseteq V$, such that every distance-preserving minor G' of G with respect to R has $|V'| \geq \Omega(k \cdot p)$. In other words, $f^*(k, \mathsf{Treewidth}(p)) \geq \Omega(pk)$.*

Proof. Fix $k \geq p$, and assume without loss of generality that p divides k. Let the graph G consist of k/p disjoint graphs $G_1, \ldots, G_{k/p}$, where each G_i is the graph from Corollary 3.7, which has p terminals, treewidth p, and every distance-preserving minor of it has at least $\Omega(p^2)$ vertices. Any distance-preserving minor of the graph G must preserve the distances between the terminals in each G_i, but this can only be achieved by a minor of the respective G_i, because the different G_i's are disjoint components. It follows that any distance-preserving minor of G must have at least $(k/p) \cdot \Omega(p^2) \geq \Omega(k \cdot p)$ vertices. ∎

5. Minors with dominating distances. Another view of a contraction of an edge (u, v) is an assymetrical contraction where one of the vertices is contracted into the other (i.e., vertex u is contracted into vertex v). The direction of contractions in the naive algorithm is easy to see—the vertex v is chosen as a vertex with degree 2 in the shortest-paths graph, and the edge (u, v) is contracted—and distances are set with respect to the vertex u. In other words, this is the contraction of the vertex v into the vertex u.

When the edge contractions are observed as such, we see that the algorithms in this paper actually satisfy a stronger property: They output a minor $G' = (V', E', \ell')$ where in effect $V' \subseteq V$ (formally, our algorithms actually map every vertex $v' \in V'$ to a specific "pre-image" $v \in V$, i.e., one of the vertices in V whose merging forms v') and the length of edges in the graph correspond to the original distance between its endpoints. An immediate result of this is that distances in G' dominate those in G, namely,

$$d_{G'}(u, v) \geq d_G(u, v) \qquad \forall u, v \in V'. \tag{5.1}$$

This additional property of distance preserving minors with dominating distances ensures sparsifications that are more closely related to the original graphs. In particular, we may view the sparsification as a simple removal of existing redundant elements, as no information is artificially added in the sparsification process. Such sparsifiers may be more easily applicable.

The following theorem proves that under this stronger property, the $O(k^4)$ bound of Theorem 2.1 is tight.

THEOREM 5.1. *For every k there exists a graph G and a set of terminals $R \subseteq V$, for which every distance-preserving minor G' where $V' \subseteq V$ and property* (5.1) *holds has $\Omega(k^4)$ vertices.*

Proof. Fix k; we construct G probabilistically as follows. Consider the unit square $[0, 1] \times [0, 1]$ in the two-dimensional Euclidean plane, and on each of its edges place terminals at $\lfloor \frac{k}{4} \rfloor$ points chosen at random. Connect by a straight line the terminals on the top edge with those on the bottom edge, and similarly connect the terminals on the right edge with those on the left edge. There are now $\Theta(k^2)$ "horizontal" lines each meeting $\Theta(k^2)$ "vertical" lines, and with probability 1 the horizontal lines intersect the vertical lines at $\Theta(k^4)$ intersection points (because the probability that three lines meet at a single point is 0). Additional intersection points might exist between pairs of horizontal lines and pairs of vertical lines. See Figure 2 for an illustration.

Let the graph G have both the terminals and the intersection points as its vertices, and their connecting line segments as its edges. Set every edge length to be the Euclidean distance between its endpoints; hence shortest-path distances in G dominate the Euclidean metric between the respective points.

Let v be an intersection point between the top-to-bottom (vertical) shortest path $\Pi_G(t_1, t_2)$ and the right-to-left (horizontal) shortest path $\Pi_G(t_3, t_4)$ in G. Let G' be a distance-preserving minor of G satisfying property (5.1) and assume toward contradiction that $v \notin V'$. Since G' is a minor of G, it can be drawn in the two-dimensional Euclidean plane such that the surviving vertices remain in the same location as they were in G and all edges are drawn inside the unit square with no crossings. Since every pair of top-to-bottom path and right-to-left path (both inside

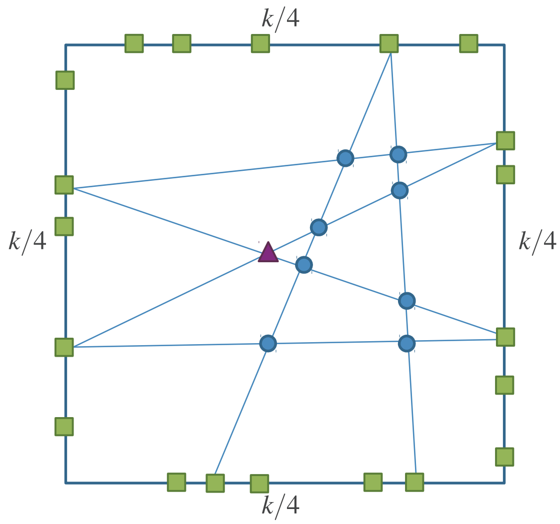

FIG. 2. *The graph constructed in the unit square, with its terminals denoted by rectangles, and showing only a few of the straight lines. Intersections between "horizontal" and "vertical" lines are denoted by a circle, and those between two "horizontal" lines are denoted by a triangle. The latter are not counted in our analysis.*

the unit square) must intersect, the shortest paths $\Pi_{G'}(t_1, t_2)$ and $\Pi_{G'}(t_3, t_4)$ intersect in some point $v' \in V'$, which must be different from v (because $v \notin V'$). But since v is the only vertex in $V \supset V'$ placed on both the straight line between t_1 and t_2, and the straight line between t_3 and t_4, one of the paths in G', say, without loss of generality $\Pi_{G'}(t_1, t_2)$, visits the point v' and goes outside of its straight line. From property (5.1) all distances in G' dominate those in G, and from the construction of G they also dominate the Euclidean metric. Hence, the length of the shortest path $\Pi_{G'}(t_1, t_2)$ is at least the sum of Euclidean distances $\|t_1 - v'\|_2 + \|v' - t_2\|_2 > \|t_1 - t_2\|_2$, making $d_{G'}(t_1, t_2) > d_G(t_1, t_2)$ in contradiction to the distance-preserving property of G'. We conclude that every intersection point between a vertical and a horizontal line in G exists also in G'; hence $|V'| \geq \Omega(k^4)$. ∎

Theorem 5.1 suggests that narrowing the gap between the current bounds $\Omega(k^2) \leq f^*(k) \leq O(k^4)$ might require, even for planar graphs, breaking away from the above paradigm and not satisfying property (5.1).

REFERENCES

[ADDJS93] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, *On sparse spanners of weighted graphs*, Discrete Comput. Geom., 9 (1993), pp. 81–100.

[BG08] A. Basu and A. Gupta, *Steiner Point Removal in Graph Metrics*, manuscript, http://www.math.ucdavis.edu/~abasu/papers/SPR.pdf (2008).

[BGHK95] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, *Approximating treewidth, pathwidth, frontsize, and shortest elimination tree*, J. Algorithms, 18 (1995), pp. 238–255.

[Bodlaender89] H. L. Bodlaender, *NC-algorithms for graphs with small treewidth*, in Proceedings of the 14th International Workshop on Graph-Theoretic Concepts in Computer Science, Springer-Verlag, 1989, pp. 1–10.

[Bodlaender06] H. L. Bodlaender, *Treewidth: Characterizations, applications, and computations*, in Proceedings of the 32nd International Conference on Graph-Theoretic Concepts in Computer Science, Springer-Verlag, 2006, pp. 1–14.

[CXKR06] T. Chan, D. Xia, G. Konjevod, and A. Richa, *A tight lower bound for the Steiner point removal problem on trees*, in Proceedings of the 9th International Workshop on Approximation, Randomization, and Combinatorial Optimization, Lecture Notes in Comput. Sci. 4110, Springer-Verlag, Berlin, 2006, pp. 70–81.

[CGNRS06] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair, *Embedding k-outerplanar graphs into ℓ_1*, SIAM J. Discret. Math., 20 (2006), pp. 119–136.

[CE06] D. Coppersmith and M. Elkin, *Sparse sourcewise and pairwise distance preservers*, SIAM J. Discrete Math., 20 (2006), pp. 463–501.

[EGKRTT10] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar, *Vertex sparsifiers: New results from old techniques*, in Proceedings of the 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization, Lecture Notes in Comput. Sci. 6302, Springer-Verlag, Berlin, 2010, pp. 152–165.

[FM95] T. Feder and R. Motwani, *Clique partitions, graph compression and speeding-up algorithms*, J. Comput. System Sci., 51 (1995), pp. 261–272.

[GH61] R. E. Gomory and T. C. Hu, *Multi-terminal network flows*, J. SIAM, 9 (1961), pp. 551–570.

[GS02] M. Grigni and P. Sissokho, *Light spanners and approximate TSP in weighted graphs with forbidden minors*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2002, pp. 852–857.

[Gupta01] A. Gupta, *Steiner points in tree metrics don't (really) help*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2001, pp. 220–227.

[HKNR98] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde, *Characterizing multiterminal flow networks and computing flows in networks of small treewidth*, J. Comput. Syst. Sci., 57 (1998), pp. 366–375.

[IS07] P. Indyk and A. Sidiropoulos, *Probabilistic embeddings of bounded genus graphs into planar graphs*, in Proceedings of the 23rd Annual Symposium on Computational Geometry, ACM, 2007, pp. 204–209.

[Klein08] P. N. Klein, *A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights*, SIAM J. Comput., 37 (2008), pp. 1926–1952.

[Kloks94] T. Kloks, *Treewidth. Computations and Approximations*, Lecture Notes in Comput. Sci. 842, Springer-Verlag, Berlin, 1994.

[KZ12] R. Krauthgamer and T. Zondiner, *Preserving terminal distances using minors*, in Proceedings of the 39th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Comput. Sci. 7391, Springer-Verlag, Berlin, 2012, pp. 594–605.

[PS89] D. Peleg and A. A. Schäffer, *Graph spanners*, J. Graph Theory, 13 (1989), pp. 99–116.

[RTZ05] L. Roditty, M. Thorup, and U. Zwick, *Deterministic constructions of approximate distance oracles and spanners*, in Proceedings of the 32nd International Conference on Automata, Languages and Programming, Springer-Verlag, Berlin, 2005, pp. 261–272.

[TZ05] M. Thorup and U. Zwick, *Approximate distance oracles*, J. ACM, 52 (2005), pp. 1–24.

[Woodruff06] D. P. Woodruff, *Lower bounds for additive spanners, emulators, and more*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2006, pp. 389–398.