Preserving Terminal Distances using Minors*

Robert Krauthgamer and Tamar Zondiner

Weizmann Institute of Science, Rehovot, Israel.
{robert.krauthgamer,tamar.zondiner}@weizmann.ac.il

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 over 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.

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 and/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

* A full version appears at http://arxiv.org/abs/1202.5675. This work was supported in part by The Israel Science Foundation (grant #452/08), by a US-Israel BSF grant #2010418, and by the Citi Foundation.

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 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. *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; 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 non-terminals, 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).

This definition may be viewed as a conceptual contribution of our paper. Indeed, our main motivation is its mathematical elegance, but let us mention one potential algorithmic application. Suppose we need to solve multiple TSP instances involving altogether relatively few vertices in a large (perhaps planar) graph; then it makes sense to reduce the graph (to a minor of it).

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. 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. Consider the naive method depicted in Algorithm 1. It is straightforward to see that these steps reduce the number of non-terminals 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 (details omitted from this version). It follows that $f^*(k)$ exists, and furthermore

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

Algorithm 1 ReduceGraphNaive (graph G, required vertices R)

(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 non-terminal 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.

Moreover, if G is a tree then G' has at most $2k-2$ vertices, and this last bound is in fact tight (attained by a complete binary tree) whenever k is a power of 2.

1.1 Our Results

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

Definition 2. *For a family $\mathcal{F}$ of graphs, define $f^*(k, \mathcal{F})$ as the minimum value 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. *Let* Planar *be the family of all planar graphs. Then*

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

Our proof of this lower bound uses a two-dimensional grid graph, which has super-constant treewidth. 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 a 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 2. *Let* $\mathsf{Treewidth}(p)$ *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.

[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.

Graph Family $\mathcal{F}$	Bounds on $f^*(k,\mathcal{F})$		
Trees	$= 2k-2$		omitted
Treewidth p	$\Omega(pk)$	$O(p^3k)$	Theorem 2
Planar Graphs	$\Omega(k^2)$	$O(k^4)$	Theorem 1
All Graphs	$\Omega(k^2)$	$O(k^4)$	Theorem 1

All our upper bounds are algorithmic and run in polynomial time. In fact, they can be achieved using the naive algorithm described above.

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 $k = O(|V|^{\frac{1}{4}})$ terminals, there exists a weighted subgraph $G' = (V,E')$ with $|E'| \leq O(|V|)$, that preserves terminal distances exactly. They also show a nearly-matching lower bound on $|E'|$.

Some compressions 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 non-terminals. Gupta [Gup01] introduced this problem and proved that for every weighted tree T and set of terminals, there exists a weighted tree T' without the non-terminals that approximates all terminal distances within a factor of 8. It was later observed that this T' is in fact a minor of T [CGN$^+$06], and that the factor 8 is tight [CXKR06]. Basu and Gupta [BG08] claimed that a constant approximation factor exists for weighted outerplanar graphs as well. It remains an open problem whether the constant factor approximation extends also to planar graphs (or excluded-minor graphs in general). Englert et al. [EGK$^+$10] 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.

2 A Lower Bound of $\Omega(k^2)$

In this section we prove Theorem 1 using an even stronger assertion: there exist planar graphs G such that every distance-preserving *planar graph* H (a planar graph with $R \subseteq V(H)$ that preserves terminal distances) has $|V(H)| \geq \Omega(k^2)$. Since any minor G' of G is planar, Theorem 1 follows.

Our proof uses a $k \times k$ grid graph with k terminals, whose edge-lengths are chosen so that terminal distances are essentially "linearly independent" of one another. We use this independence to prove that no distance-preserving minor G' can have a small vertex-separator. Since G' is planar, we can apply the planar separator theorem [LT79], and obtain the desired lower bound.

Theorem 3. *For every $k \in \mathbb{N}$ there exists a planar graph $G = (V, E, \ell)$ (in particular, the $k \times k$ grid) and k terminals $R \subseteq V$, such that every distance-preserving* planar graph *$G' = (V', E', \ell')$ has $\Omega(k^2)$ vertices. In particular, $f^*(k, \text{Planar}) \geq \Omega(k^2)$.*

Proof. For simplicity we shall assume that k is even. Consider a grid graph G of size $k \times k$ with vertices (x, y) for $x, y \in [0, k-1]$. Let the length function ℓ be such that the length of all horizontal edges $((x, y), (x+1, y))$ is 1, and the length of each vertical edge $((x, y), (x, y+1))$ is $1 + \frac{1}{2^{x^2} \cdot k}$. Let $R_1 = \{(0, y) : y \in [0, \frac{k}{2} - 1]\}$, and $R_2 = \{(x, x) : x \in [\frac{k}{2}, k-1]\}$. Let the terminals in the graph be $R = R_1 \cup R_2$, so $|R| = k$. See Figure 1 for illustration.

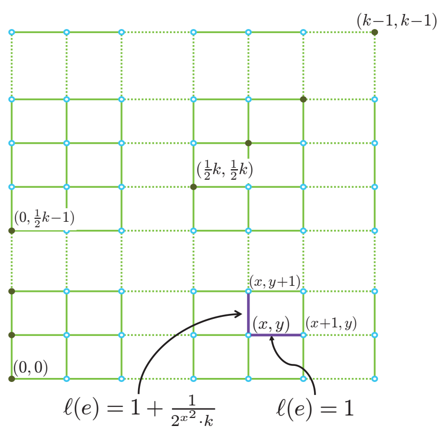

Fig. 1. A grid graph G and terminals R.

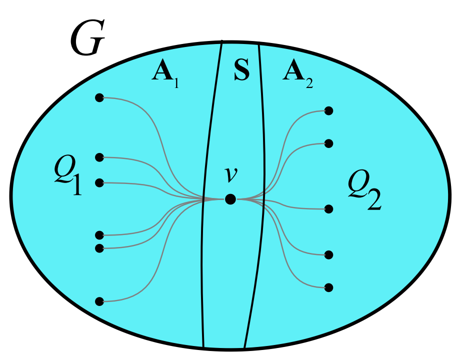

Fig. 2. Terminals on different sides connected by paths going through $v \in S$.

It is easy to see that the shortest-path between a vertex $(0, y) \in R_1$ and a vertex $(x, x) \in R_2$ includes exactly x horizontal edges and $x - y$ vertical edges. Indeed, such paths have length smaller than $x + (x - y)(1 + \frac{1}{k}) \leq 2x - y + 1$. Any other path between these vertices will have length greater than $2x - y + 2$.

Furthermore, the shortest path with x horizontal edges and $x-y$ vertical edges starting at vertex $(0,y)$ makes horizontal steps before vertical steps, since the vertical edge-lengths decrease as x increases, hence

$$d_G((0,y),(x,x)) = 2x - y + \frac{x-y}{2^{x^2}\cdot k}. \tag{1}$$

Assume towards contradiction that there exists a planar graph G' with less than $\frac{k^2}{1600}$ vertices that preserves terminal distances exactly. Since G' is planar, by the weighted version of the planar separator theorem by Lipton and Tarjan [LT79] with vertex-weight 1 on terminals and 0 on non-terminals, there exists a partitioning of V' into three sets A_1, S, and A_2 such that $w(S) \le |S| \le 2.5 \cdot \sqrt{\frac{k^2}{1600}} < \frac{3k}{40}$, each of A_1 and A_2 has at most $\frac{2k}{3}$ terminals, and there are no edges going between A_1 and A_2. Hence, for $i \in \{1,2\}$ it holds that $w(A_i \cup S) \ge k/3$ and $w(A_i) \ge \frac{k}{3} - \frac{3k}{40} > \frac{k}{4}$.

Without loss of generality, we claim that $A_1 \cap R_1$ and $A_2 \cap R_2$ each have $\Theta(k)$ terminals. To see this, suppose without loss of generality that A_1 is the heavier of the two sets (i.e. $w(A_1) \ge \frac{k}{2} - \frac{3k}{40}$ and $\frac{k}{4} \le w(A_2) \le \frac{k}{2}$). Suppose also that $w(A_2 \cap R_2) \ge w(A_2 \cap R_1)$. Then $w(A_2 \cap R_2) \ge \frac{k}{8}$, and $w(A_2 \cap R_1) \le \frac{1}{2} \cdot w(A_2) \le \frac{k}{4}$, implying that $w(A_1 \cap R_1) \ge w(R_1) - (w(R_1 \cap A_2) + w(R_1 \cap S)) \ge \frac{k}{2} - (\frac{k}{4} + \frac{3k}{40}) = \frac{k}{5}$. In conclusion, without loss of generality it holds that $w(A_1 \cap R_1) \ge \frac{k}{5}$ and $w(A_2 \cap R_2) \ge \frac{k}{8}$. Let $Q_1 \subseteq A_1 \cap R_1$ and $Q_2 \subseteq A_2 \cap R_2$ be two sets with the exact sizes $\frac{k}{5}$ and $\frac{k}{8}$.

Every path between a terminal in Q_1 and a terminal in Q_2 goes through at least one vertex of the separator S. Overall, the vertices in the separator participate in $\frac{k}{8} \times \frac{k}{5}$ paths between Q_1 and Q_2. See Figure 2 for illustration.

We will need the following lemma, which is proved below.

Lemma 1. *Let G', S, Q_1 and Q_2 be as described above. Then every vertex $v \in S$ participates in at most $|Q_1| + |Q_2| = \frac{k}{5} + \frac{k}{8}$ shortest paths between Q_1 and Q_2.*

Applying Lemma 1 to every vertex in S, at most $\frac{3k}{40} \cdot \frac{13k}{40} = \frac{39k^2}{1600} < \frac{k^2}{40}$ shortest paths between Q_1 and Q_2 go through S, which contradicts the fact that all $\frac{k}{8} \cdot \frac{k}{5} = \frac{k^2}{40}$ shortest-paths between Q_1 and Q_2 in G' go through the separator, and proves Theorem 3. ☐

Proof (of Lemma 1). Define a bipartite graph H on the sets Q_1 and Q_2, with an edge between $(0,y) \in Q_1$ and $(x,x) \in Q_2$ whenever a shortest path in G' between $(0,y)$ and (x,x) uses the vertex v. We shall show that H does not contain an even-length cycle. Since H is bipartite, it contains no odd-length cycles either, making H a forest with $|E(H)| < |Q_1| + |Q_2| = \frac{k}{5} + \frac{k}{8}$, thereby proving the lemma.

Let us consider a potential $2s$-length (simple) cycle in H on the vertices $(0,y_1)$, (x_1,x_1), $(0,y_2)$, (x_2,x_2), ..., $(0,y_s)$, (x_s,x_s) (in that order), for particular $(0,y_i) \in Q_1$ and $(x_i,x_i) \in Q_2$. Every edge $((0,y),(x,x)) \in E(H)$ represents a shortest path in G' that uses v, thus

$$d_G((0,y),(x,x)) = d_{G'}((0,y),v) + d_{G'}(v,(x,x)). \tag{2}$$

If the above cycle exists in H, then the following equalities hold (by convention, let $y_{s+1} = y_1$). Essentially, we get that the sum of distances corresponding to "odd-numbered" edges in the cycle equals the one corresponding to "even-numbered" edges in the cycle.

$$\begin{aligned}\sum_{i=1}^{s} d_G((0,y_i),(x_i,x_i)) &\overset{(2)}{=} \sum_{i=1}^{s} d_{G'}((0,y_i),v) + \sum_{i=1}^{s} d_{G'}(v,(x_i,x_i)) \\ &= \sum_{i=1}^{s} d_{G'}(v,(0,y_{i+1})) + \sum_{i=1}^{s} d_{G'}((x_i,x_i),v) \\ &\overset{(2)}{=} \sum_{i=1}^{s} d_G((x_i,x_i),(0,y_{i+1})).\end{aligned}$$

Plugging in the distances as described in (1) and simplifying, we obtain

$$\sum_{i=1}^{s}(2x_i - y_i + (x_i - y_i)\cdot\frac{1}{2^{x_i^2}\cdot k}) = \sum_{i=1}^{s}(2x_i - y_{i+1} + (x_i - y_{i+1})\cdot\frac{1}{2^{x_i^2}\cdot k}),$$

or equivalently,

$$\sum_{i=1}^{s}\frac{y_i}{2^{x_i^2}} = \sum_{i=1}^{s}\frac{y_{i+1}}{2^{x_i^2}}$$

Suppose without loss of generality that $x_1 = \min\{x_i : i \in [1,s]\}$ (otherwise we can rotate the notations along the cycle), and that $y_1 > y_2$ (otherwise we can change the orientation of the cycle). Then we obtain

$$\frac{y_1 - y_2}{2^{x_1^2}} = \sum_{i=2}^{s}\frac{y_{i+1} - y_i}{2^{x_i^2}}.$$

However, since $y_1 > y_2$, the lefthand side is at least $\frac{1}{2^{x_1^2}}$, whereas the righthand side is $\sum_{i=2}^{s}\frac{y_{i+1}-y_i}{2^{x_i^2}} \le (s-1)\cdot\frac{k}{2^{(x_1+1)^2}} \le \frac{k^2}{2^{(x_1+1)^2}}$. Therefore it must hold that $2^{2x_1+1} \le k^2$. Since $x_1 \ge \frac{k}{2}$, this inequality does not hold. Hence, for all s, no cycle of size $2s$ exists in H, completing the proof of Lemma 1. □

3 $\Theta(k)$ Bounds for Constant Treewidth Graphs

In this section we prove Theorem 2, which bounds $f^*(k, \mathsf{Treewidth}(p))$. The upper and the lower bound are proved separately in Theorems 4 and 5 below.

3.1 An Upper Bound of $O(p^3k)$

Theorem 4. *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 $|V'| \le O(p^3k)$. In other words, $f^*(k, \mathsf{Treewidth}(p)) \le O(p^3k)$.*

The graph G' can in fact be computed in time polynomial in $|V|$ (see Remark 1).

Without loss of generality, we may assume that $k \geq p$, since otherwise the $O(k^4)$ bound mentioned in the introduction applies. To prove Theorem 4 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 execute ReduceGraphNaive (Algorithm 1) 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 — and we get many small distance-preserving minors; these are then combined into a distance-preserving minor G' of the original graph G.

Proof (of Theorem 4). 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 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 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 $G[A_2 \cup S]$. 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 $i \in \{1, 2\}$. Altogether, the shortest path in G is preserved in $\hat{G}$. It is easy to see that shorter paths will never be created, as these too can be split into subpaths such that the length of each subpath is preserved. Hence, $\hat{G}$ is a distance-preserving minor of G. □

The graph G has bounded treewidth p, hence 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 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 [Bod89]) 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.

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: **return** $(\hat{G}_1^1 \cup \hat{G}_1^2) \cup (\hat{G}_2^1 \cup \hat{G}_2^2)$.

See Figure 3 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 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 $|R_{b_i}| \leq \frac{2}{3}|R_a|$. 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 3. *For every $a \in V(T)$, the number of boundary vertices $|B_a| < 6p$.*

Proof (of Lemma 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

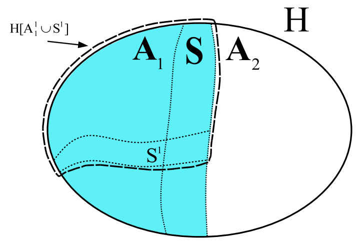

Fig. 3. The separators S (from line 3) and S^1 (from line 7), and the subgraph $H[A_1^1 \cup S^1]$ to be processed recursively (in line 11).

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, $|S^i| \leq p$, $|A_i^j| \leq \frac{2}{3} \cdot |B_a \cup S| \leq \frac{2}{3} \cdot 7p = \frac{14}{3}p$, and so $|A_i^j \cup S^i| \leq \frac{14}{3}p + p < 6p$. 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. *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 $|R_{b_1}|, |R_{b_2}| \geq p$.*

Proof (of Lemma 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 $|R_a \cup B_a| \geq 18p$. Since Lemma 3 states that $|B_a| \leq 6p$ it must holds that $|R_a| \geq 12p$.

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 $|R_a \cap (A_i \cup S)| \geq \frac{1}{3}|R_a| = 4p$ for $i = 1, 2$. Since $|S| \leq p$ it must holds that $|(R_a \setminus S) \cap A_i| = |(R_a \cap (A_i \cup S)) \setminus S| \geq 3p$. 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 $|R^i \cap A_i^1| \geq \frac{1}{2}2p = p$. 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 $|R_{b_i}| \geq p$ are made. □

The following observation is immediate from Lemma 3.

Observation 1. *Every node $a \in V(T)$ such that $|R_a| < p$ has $|R_a \cup B_a| \leq 7p$, thus 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 1. From Lemma 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, invocations are performed only by internal vertices in T'. Each internal vertex has 4 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. Using Lemma 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. □

Remark 1. Every action (edge or vertex removals, as well as edge contractions) taken by ReduceGraphTW, is actually performed during a call to ReduceGraphNaive, and an equivalent action to it would have been taken in executing the naive algorithm directly on G with respect to terminals R. Therefore, the naive algorithm returns distance-preserving minors of size $O(k \cdot p^3)$ to any graph with treewidth p. (When $p > k$ this statement holds by the $O(k^4)$ bound.)

3.2 A Lower Bound of $\Omega(pk)$

Theorem 5. *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. Consider the bound shown in Theorem 3. The graph used to obtain this bound is a $k \times k$ grid, and has treewidth k. The following corollary holds.

Corollary 1. *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 $|V'| \geq \Omega(p^2)$.*

Let the graph G consist of $\frac{k}{p}$ disjoint graphs G_i with p terminals, treewidth p, and distance-preserving minors with $|V'| \geq \Omega(p^2)$ as guaranteed by Corollary 1. Any distance-preserving minor of the graph G must preserve (in disjoint components) the distances between the terminals in each G_i. The graph G has k terminals, treewidth p, and any distance-preserving minor of it has $|V'| \geq \Omega(k \cdot p)$, thus proving Theorem 5. □

4 Concluding Remarks

The algorithms mentioned in this paper (including the naive one) actually satisfy a stronger property: They output a minor $G' = (V', E', \ell')$ where $V' \subset V$

(meaning that every vertex in G' can be mapped back to a vertex in G) and

$$d_{G'}(u,v) \geq d_G(u,v) \qquad \forall u,v \in V'. \tag{3}$$

However, it is not hard to construct instances G (say, using Euclidean distances between random points in the plane, which yields in particular a planar graph), for which every distance-preserving minor G' satisfying the stronger property (3) must have $\Omega(k^4)$ vertices. Therefore, 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.

References

[BG08] A. Basu and A. Gupta. Steiner point removal in graph metrics. Unpublished Manuscript, available from `http://www.math.ucdavis.edu/~abasu/papers/SPR.pdf`, 2008.

[Bod89] H. L. Bodlaender. NC-algorithms for graphs with small treewidth. In *14th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 1–10. Springer-Verlag, 1989.

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

[CGN+06] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Embedding k-outerplanar graphs into ℓ_1. *SIAM J. Discret. Math.*, 20(1):119–136, 2006.

[CXKR06] T. Chan, D. Xia, G. Konjevod, and A. Richa. A tight lower bound for the Steiner point removal problem on trees. In *9th International Workshop on Approximation, Randomization, and Combinatorial Optimization*, volume 4110 of *Lecture Notes in Computer Science*, pages 70–81. Springer, 2006.

[EGK+10] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. In *13th International Workshop on Approximation, Randomization, and Combinatorial Optimization*, volume 6302 of *Lecture Notes in Computer Science*, pages 152–165. Springer, 2010.

[FM95] T. Feder and R. Motwani. Clique partitions, graph compression and speeding-up algorithms. *J. Comput. Syst. Sci.*, 51(2):261–272, 1995.

[GH61] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9:551–570, 1961.

[Gup01] A. Gupta. Steiner points in tree metrics don't (really) help. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 220–227. SIAM, 2001.

[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.*, 57:366–375, 1998.

[LT79] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36(2):177–189, 1979.

[PS89] D. Peleg and A. A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.

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