Degree-3 Treewidth Sparsifiers*

Chandra Chekuri† Julia Chuzhoy‡

Abstract

We study treewidth sparsifiers. Informally, given a graph G of treewidth k, a treewidth sparsifier H is a minor of G, whose treewidth is close to k, $|V(H)|$ is small, and the maximum vertex degree in H is bounded. Treewidth sparsifiers of degree 3 are of particular interest, as routing on node-disjoint paths, and computing minors seems easier in sub-cubic graphs than in general graphs. In this paper we describe an algorithm that, given a graph G of treewidth k, computes a topological minor H of G such that (i) the treewidth of H is $\Omega(k/\text{polylog(k)})$; (ii) $|V(H)| = O(k^4)$; and (iii) the maximum vertex degree in H is 3. The running time of the algorithm is polynomial in $|V(G)|$ and k. Our result is in contrast to the known fact that unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, treewidth does not admit polynomial-size kernels. One of our key technical tools, which is of independent interest, is a construction of a small minor that preserves node-disjoint routability between two pairs of vertex subsets. This is closely related to the open question of computing small good-quality vertex-cut sparsifiers that are also minors of the original graph.

1 Introduction

Given a large graph G, the goal in graph sparsification is to compute a "small" graph H that retains, exactly or approximately, some key properties of G. Two such standard regimes are when $V(H) = V(G)$ but H is a sparse graph, or when $|V(H)| \ll |V(G)|$. Sparsifiers for basic properties such as connectivity, distances, cuts and flows have been extensively studied. For instance, cut sparsifiers were introduced by Benczur and Karger [4], and were more recently generalized to spectral sparsifiers [3], and to cut and flow sparsifiers for vertex subsets [32, 28]. Graph sparsifiers are closely related to the notion of kernelization used in fixed-parameter tractable algorithms, where an input instance is first reduced to a much smaller instance (called a kernel), whose size is ideally polynomial in the parameter k, and then the problem is solved on the smaller instance. Sparsification and sparse representations are also of great importance for other objects such as signals, matrices, and geometric objects to name just a few.

We say that a graph H is a *strong* sparsifier for the given graph G, if additionally H is a minor of G. Strong sparsifiers are of particular interest, since they retain some of the structure of G. For example, if H contains some graph H' as a minor, then so does G; a collection $\mathcal{P}$ of disjoint paths (or cycles) in H immediately translates to a collection of disjoint paths (or cycles) in G, and so on.

In this paper we study sparsifiers for *treewidth*, a fundamental graph parameter with a wide variety of applications in graph theory and algorithms. The treewidth of a graph $G = (V, E)$ is typically defined via tree decompositions. A tree-decomposition of G consists of a tree $T = (V(T), E(T))$ and a collection of vertex subsets $\{X_v \subseteq V\}_{v \in V(T)}$ called bags, such that: (i) for each edge $(a, b) \in E$, there is some node $v \in V(T)$ with both $a, b \in X_v$ and (ii) for each vertex $a \in V$, the set of all nodes of T whose bags contain a form a non-empty connected subtree of T. The *width* of a given tree decomposition is $\max_{v \in V(T)} |X_v| - 1$, and the treewidth of a graph G, denoted by $\text{tw}(G)$, is the width of a minimum-width tree decomposition for G. Treewidth is known to be NP-hard to compute [2]. The best known polynomial-time approximation algorithm, given a graph G of treewidth k, computes a tree decomposition of width $O(k\sqrt{\log k})$ [18]. It is also known that treewidth is fixed-parameter-tractable [5]: for every fixed k, there is a linear-time algorithm, that, given G, decides whether $\text{tw}(G) \le k$; the dependence of the running time on k is exponential in $\text{poly}(k)$. There are many important results on the structure of large-treewidth graphs. Perhaps the most well-known of these is the Grid-Minor Theorem of Robertson and Seymour that we discuss in more detail later.

Informally, graph H is a treewidth sparsifier for a given graph G, if H is sparse, $|V(H)|$ is small, and $\text{tw}(H)$ is (approximately) the same as $\text{tw}(G)$. For H to be useful as a replacement for G, it needs to be a strong sparsifier

*A full version of the paper is available on the ArXiv [10].

†Dept. of Computer Science, University of Illinois, Urbana, IL 61801. `chekuri@illinois.edu`. Supported in part by NSF grant CCF-1319376.

‡Toyota Technological Institute, Chicago, IL 60637. Email: `cjulia@ttic.edu`. Supported in part by NSF grant CCF-1318242.

— that is, H should be a minor of G[1]. The notion of treewidth sparsifiers is closely related to the notion of kernels for treewidth. A polynomial kernel for treewidth is a map f, that, given an instance (G, k), returns an instance (G', k'), with the property that $\mathrm{tw}(G) \le k$ iff $\mathrm{tw}(G') \le k'$, while ensuring that the size of the graph G' is polynomial in k. Unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ there is no polynomial kernel for treewidth which follows from the results of Bodlaender et al. [6] and Drucker [16]. Super-linear lower bounds for more general forms of kernelization are also known [21].

Our main result shows that if one is willing to settle for a poly-logarithmic factor approximation in the treewidth, then there exist sparsifiers with very strong properties. To state our main result we need a definition. A graph H is a topological minor of G if H is obtained from G by edge and node deletions, and by suppressing degree-2 nodes[2]. Equivalently, H is a topological minor of G iff a subdivision of H is a subgraph of G. Our main result is summarized in the following theorem.

Theorem 1.1. *There is a randomized algorithm, that, given a graph G of treewidth at least k, with high probability computes a topological minor H of G, such that:*

- *the treewidth of H is $\Omega(k/\operatorname{poly}\log k)$;*
- *the maximum vertex degree in H is 3; and*
- *$|V(H)| = O(k^4)$.*

The running time of the algorithm is polynomial in $|V(G)|$ and k.

Our result is close to optimal: degree 3 cannot be reduced, and the best one can hope for in terms of the size of the sparsifier is $O(k^2/\operatorname{poly}\log k)$ (when G is a $k \times k$ grid). We also recall that the best currently known polynomial-time approximation algorithm can only certify treewidth to within an $O(\sqrt{\log k})$-factor. We conjecture a strengthening of the theorem to almost optimal parameters.

[1]Note that if all we wanted is a graph H that has similar treewidth as G then it suffices to (approximately) compute $\mathrm{tw}(G)$ and let H be any graph from a well-known class such as grids, cliques or expanders with the same treewidth.

[2]Note that H is a minor of G if it can be obtained by edge and node deletions and edge contractions. A minor H of a graph G need not be a topological minor G, however, if the maximum vertex degree in H is at most 3, then H is also a topological minor of G.

Conjecture 1.1. *For every graph G with treewidth at least k, there* exists *a topological minor H of G such that* $\mathrm{tw}(H) = \Omega(k/\operatorname{poly}\log k)$, $|V(H)| = O(k^2)$ *and maximum vertex degree in H is* 3.

The existence of sparsifiers of size $\operatorname{poly}(k)$ that preserve the treewidth to within a constant factor remains a very interesting open question.

1.1 Treewidth Sparsifiers and Grid Minors

A fundamental result in Graph Minor Theory is the Grid-Minor Theorem of Robertson and Seymour [37]. The theorem states that there is an integer-valued function f, such that any graph G with treewidth at least $f(g)$ contains a $g \times g$ grid as a minor. The theorem is equivalent to showing that $\mathrm{tw}(G) \ge f(g)$ implies that G contains a *wall* of height and width $\Theta(g)$ as a subgraph; see Figure 1.

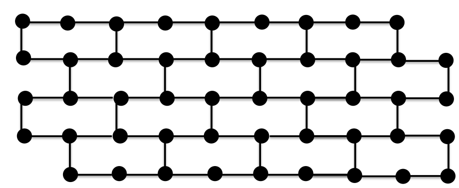

Figure 1: An elementary wall of height and width 5. A wall is a subdivision of an elementary wall.

We observe that a wall has maximum vertex degree 3. Thus, one way to obtain a degree-3 treewidth sparsifier is via the Grid-Minor Theorem. The original proof of Robertson and Seymour [37] showed the existence of f with an iterated exponential dependence on g. Very recently, the first polynomial bound on f was shown in [9]: namely, every graph of treewidth k contains a wall of size k^δ as a topological minor, where $\delta = 1/98 - o(1)$. This result implies a degree-3 treewidth sparsifier, whose treewidth is $k^{1/98-o(1)}$. In contrast, the sparsifier from Theorem 1.1 has treewidth $\Omega(k/\text{polylog(k)})$. Moreover, there are graphs with treewidth k, such that the size of the largest wall they contain is $O(\sqrt{k/\log k})$ [36]. Therefore, one cannot hope to obtain small sparsifiers that preserve treewidth to within polylogarithmic factors via the Grid-Minor Theorem. Our construction bypasses this limitation.

One of our motivations for studying treewidth sparsifiers is improving the bounds for the Grid-Minor Theorem. Theorem 1.1 allows us to focus on subcubic graphs with the additional property that $|V(G)|$ is polynomial in $\mathrm{tw}(G)$. Degree-3 sparsifiers have particular advantages: in such a graph, for several applications of interest, one can replace node-disjoint routing with the easier edge-disjoint routing. We anticipate that using Theorem 1.1 as a starting point, the bounds on the Grid-Minor Theorem from [9] can be improved. We also mention that the fact that $|V(H)| = \text{poly}(k)$ simplifies some technical parts in the current proof of [9].

A related application is to the notion of graph immersions (see [38, 39]). A graph G admits a strong immersion of a graph H iff there is an injective mapping $\tau : V(H) \to V(G)$ and a mapping $\pi : E(H) \to \mathcal{P}_G$, where $\mathcal{P}_G$ is a set of paths in G, such that (i) for each $f = (a, b) \in E(H)$ the path $\pi(f)$ connects $\tau(a)$ and $\tau(b)$; (ii) for any two edges $f, f' \in E(H)$ the paths $\pi(f)$ and $\pi(f')$ are edge-disjoint; and (iii) for every $f \in E(H)$ the path $\pi(f)$ intersects $\tau(V(H))$ only at its endpoints. Note that G admits H as a topological minor if additionally the paths $\pi(f)$ and $\pi(f')$ are internally *node*-disjoint for any distinct pair $f, f' \in E(H)$. If G is a sub-cubic graph, then G contains H as a topological minor iff G contains H as a strong immersion. Therefore, G contains a wall W iff it contains it as an immersion. In recent work, Wollan [39] defined the notion of tree-cut width of a graph and showed, using the Grid-Minor Theorem, that there is a function g, such that every graph with tree-cut width at least $g(r)$ admits an r-wall as a weak immersion. Motivated by this connection, he raised the question of the existence of degree-3 treewidth sparsifiers. Theorem 1.1 answers his question (Question 18 in [39]) in a near-optimal fashion and we refer the reader to [39] for the quantitative and qualitative implications to immersions.

Our result can be viewed as providing an approximate kernel for treewidth, and we hope that it will find applications in preprocessing graphs for fixed-parameter tractable (FPT) algorithms, and in constructive aspects of Erdos-Pösa type theorems.

We now briefly discuss our techniques. We use a combinatorial object, called a path-of-sets system, that was defined in [9] (see also Figure 2). Using the construction of the Path-of-Sets system from [9], together with the Cut-Matching Game of Khandekar, Rao and Vazirani [24], we can immediately obtain a strong degree-4 treewidth sparsifier H, with $\text{tw}(H) = \Omega(k/\text{polylog(k)})$. However, the size of $V(H)$ can be arbitrarily large. Our main technical contribution is two-fold. First, we lower the degree of the sparsifier to 3, by carefully sub-sampling the edges of H. Second, we reduce the size of the sparsifier to $\text{poly}(k)$. For the second part, we crucially need a new technical ingredient, that is related to strong vertex-cut sparsifiers, that we discuss below.

1.2 Sparsifiers Preserving Vertex Cuts

Suppose we are given any graph $G = (V, E)$ and a pair $S, T \subseteq V$ of vertex subsets, containing k vertices each. We say that the pair (S, T) is *routable* in G iff there is a set $\mathcal{P}$ of k disjoint paths connecting the vertices of S to the vertices of T in G, and we say that the set $\mathcal{P}$ of paths *routes* the pair (S, T). Assume now that we have two pairs of vertex subsets: S_1, T_1, containing k_1 vertices each, and S_2, T_2 containing k_2 vertices each. We say that both pairs (S_1, T_1), (S_2, T_2) are *separately routable*, or just routable, in G iff there is a set $\mathcal{P}$ of paths routing (S_1, T_1), and there is a set $\mathcal{Q}$ paths routing (S_2, T_2) in G. Note that a vertex of G may belong to a path in $\mathcal{P}$ and a path in $\mathcal{Q}$. Our second main result is summarized in the following theorem.

THEOREM 1.2. *Assume that we are given a graph G, two sets $S_1, T_1 \subseteq V(G)$ of k_1 vertices each, and two sets $S_2, T_2 \subseteq V(G)$ of k_2 vertices each, such that $k_1 \geq k_2$, and the pairs (S_1, T_1) and (S_2, T_2) are (separately) routable in G. Then there are two sets $\mathcal{P}, \mathcal{Q}$ of paths routing (S_1, T_1) and (S_2, T_2) respectively, such that, if H is the graph obtained by the union of the paths in $\mathcal{P}$ and $\mathcal{Q}$, then $\tau(H) \leq 8k_1^4 + 8k_1$, where $\tau(H)$ is the number of nodes of degree more than two in H. Moreover, we can find $\mathcal{P}$ and $\mathcal{Q}$ in time polynomial in n and k_1.*

The preceding theorem gives an upper bound on the size of a topological minor of G that preserves the vertex connectivity between S_1, T_1 and S_2, T_2. There are results in the literature on reduction operations that preserve edge connectivity [29, 30] (and also element connectivity [20, 12]), however no such nice operations are available for preserving vertex connectivity. We briefly discuss some related work on cut sparsifiers and an open problem on a generalization of Theorem 1.2 that would yield strong sparsifiers that preserve vertex cuts.

There has been a large amount of work in the recent past on graph sparsifiers that preserve cuts and flows for subsets of vertices [32, 28, 8, 31, 17, 13]. We discuss some closely related work. Given an edge-capacitated graph G and a terminal set $\mathcal{T} \subseteq V(G)$, a graph H is a quality-q cut-sparsifier for $\mathcal{T}$ if (i) $\mathcal{T} \subseteq V(H)$ and (ii) for any partition (A, B) of $\mathcal{T}$, $\text{MinCut}_G(A, B) \leq \text{MinCut}_H(A, B) \leq q \,\text{MinCut}_G(A, B)$ where $\text{MinCut}_F(A, B)$ is the minimum edge-cut separating A from B in a graph F. Quality-1 sparsifiers have also been called *mimicking networks* in prior work [19, 26, 23, 7]. Leighton and Moitra [28] have shown that for any graph G, there is a quality-q sparsifier H for G with $q = O(\log k/ \log\log k)$ and $V(H) = \mathcal{T}$ (here $k = |\mathcal{T}|$); in other words the sparsifier does not use any non-terminal (or Steiner) vertices. There are instances on which the best quality one can achieve is $\Omega(\sqrt{\log k})$ if H does not have Steiner vertices [31]. Even a relatively small number of Steiner vertices can help substantially in improving the quality of the sparsifier as shown in [13].

To simplify the discussion, we restrict our attention to the case where the terminals in $\mathcal{T}$ have degree 1 and all edge capacities are 1. In this case constant quality cut-sparsifiers are known with $V(H) = O(k^3)$ [13, 25]. The result of Kratsch and Wahlström [25], in fact, applies in the more general setting of vertex-cuts, and yields a quality-1 sparsifier; we call such a sparsifier a vertex-cut sparsifier to distinguish it from an edge-cut sparsifier.

However, the sparsifer of [25] is *not* a minor of the original graph G. Sparsifiers that are minors of the original graph have an advantage that they allow flows (fractional or integral) and minors in the sparsifier to be transferred back to the original graph G without any loss. Theorem 1.2 gives us a small-sized minor that preserves the vertex connectivity between two pairs of vertex subsets. A natural open question is to generalize this result to a larger number of pairs of vertex subsets.

QUESTION 1.2. *Assume that we are given a graph G, and h pairs of vertex subsets $(S_1, T_1), \ldots, (S_h, T_h)$, such that for each i: (1) $S_i, T_i \subseteq V(G)$, (2) $|S_i| = |T_i| = k_i \leq k$, and (3) (S_i, T_i) are routable in G. What is the smallest function $f(k, h)$, such that, given any graph G and $(S_1, T_1), \ldots, (S_h, T_h)$ as above, there is always a (topological) minor H of G with the property that each (S_i, T_i) is routable in H and $|V(H)| \leq f(k, h)$?*

The case when $h = \text{polylog(k)}$ is of particular interest. We believe that a bound on $f(k, h)$ from the preceding question can be used to obtain a vertex-cut sparsifier H for any graph G and a set $\mathcal{T}$ of k terminals, such that H is a minor of G, $|V(H)| \leq f(\text{poly}\, k, h)$ for $h = \text{poly} \log k$, and the quality of H is polylog(k).

Organization We prove Theorem 1.2 in Section 2. Section 3 provides the necessary background on treewidth and the path-of-sets system. Theorem 1.1 is proved in two steps. Section 4 gives the proof of a weaker result, a degree-4 sparsifier. Section 5 gives the proof for the degree-3 sparsifier.

In this extended abstract we omit some proofs. For these we refer the reader to a full version of the paper [10].

2 Routing Two Pairs of Vertex Subsets

In this section we prove Theorem 1.2. Recall that a graph H is a *minor* of a graph G, iff H can be obtained from G by a series of edge deletion, vertex deletion, and edge contraction operations. Equivalently, H is a minor of G iff there is a map $f : V(H) \to 2^{V(G)}$ assigning to each vertex $v \in V(H)$ a subset $f(v)$ of vertices of G, such that: (a) for each $v \in V(H)$, the sub-graph of G induced by $f(v)$ is connected; (b) if $u, v \in V(H)$ and $u \neq v$, then $f(u) \cap f(v) = \emptyset$; and (c) for each edge $e = (u, v) \in E(H)$, there is an edge in $E(G)$ with one endpoint in $f(v)$ and the other endpoint in $f(u)$. A map f satisfying these conditions is called *a model of H in G*. Given any subset $X \subseteq V$ of vertices of G, we say that H is an X-respecting minor of G, iff $X \subseteq V(H)$. More formally, there is a model f of H, where for each vertex $x \in X$, there is a distinct vertex $v_x \in V(H)$ with $f(v_x) = \{x\}$. For each $x \in X$, we will usually identify such vertex v_x with x. In particular, every subset $S \subseteq X$ of vertices of X corresponds to a subset $S' = \{v_x \mid x \in X\}$ of vertices in H, and we will not distinguish between S and S'.

Assume that we are given a graph G and two pairs (S'_1, T'_1), (S'_2, T'_2) of vertex subsets, with $|S'_1| = |T'_1|$ and $|S'_2| = |T'_2|$, that are separately routable in G. We say that a minor H of G is (S'_1, T'_1, S'_2, T'_2)-good, iff H is an X-respecting minor for $X = S'_1 \cup S'_2 \cup T'_1 \cup T'_2$, and (S'_1, T'_1), (S'_2, T'_2) are each routable in H. We say that it is (S'_1, T'_1, S'_2, T'_2)-minimal, iff it is (S'_1, T'_1, S'_2, T'_2)-good, and for every edge e of H, both the graph obtained from H by deleting e, and the graph obtained from H by contracting e, are not (S'_1, T'_1, S'_2, T'_2)-good. The main result of this section is the following theorem.

THEOREM 2.1. *Assume that we are given a graph G, and sets $S'_1, T'_1, S'_2, T'_2 \subseteq V(G)$ of k vertices each, such that the pairs (S'_1, T'_1) and (S'_2, T'_2) are (separately) routable in G. Assume further that vertices in S'_1, T'_1, S'_2, T'_2 are distinct, and have degree* 1 *each in G. Let H be any (S'_1, T'_1, S'_2, T'_2)-minimal minor of G. Then $|V(H)| \leq 4k^4 + 4k$.*

Theorem 1.2 easily follows from Theorem 2.1; see [10].

In the rest of this section, we focus on the proof of Theorem 2.1. For simplicity, we denote S'_1, S'_2, T'_1, T'_2 by S_1, S_2, T_1 and T_2, respectively. Let H be a (S_1, T_1, S_2, T_2)-minimal minor of G. Let $\mathcal{R}$ be a set of paths routing (S_1, T_1) in H. We will often refer to the paths in $\mathcal{R}$ as *red paths*, and we will think of these paths as directed from S_1 towards T_1 (even though in general the graph is undirected). Similarly, let $\mathcal{B}$ be the set of paths routing (S_2, T_2) in H. We refer to the paths in $\mathcal{B}$ as *blue paths*, and view them as directed from S_2 to T_2. Notice that a vertex in $S_1 \cup T_1$ cannot participate in a blue path, since its degree is 1, and all vertices in S_1, T_1, S_2, T_2 are distinct. Similarly, a vertex in $S_2 \cup T_2$ cannot participate in a red path. An edge $e \in E(H)$ may belong to a red path, or to a blue path, but not both, since otherwise we could contract e and obtain a minor that is still (S_1, T_1, S_2, T_2)-good, contradicting the minimality of H; this is possible since e is not incident on $S_1 \cup S_2 \cup T_1 \cup T_2$. The edges that belong to the paths in $\mathcal{R}$ are called *red edges*, and the edges that belong to the paths in $\mathcal{B}$ are called *blue edges*. From the minimality of H, every edge is either red or blue. We will refer to the vertices in $S_1 \cup S_2 \cup T_1 \cup T_2$ as the terminals of H. From the minimality of H, every non-terminal vertex belongs to one red path and one blue path, and is incident on exactly two red edges and exactly two blue edges. Assume now that there is another set $\mathcal{R}' \neq \mathcal{R}$ of paths in H routing (S_1, T_1). Then there must be some red edge in H that does not belong to any path of $\mathcal{R}'$, contradicting the minimality of H. Therefore, $\mathcal{R}$ is the unique set of paths routing (S_1, T_1) in H, and similarly, $\mathcal{B}$ is the unique set of paths routing (S_2, T_2) in H. We prove the following theorem.

THEOREM 2.2. *We can efficiently compute an assignment of labels in $L = \{\ell_1, \ell_2, \ldots, \ell_{2k}\}$ to the vertices of $V(H)$, such that each vertex in $V(H)$ is assigned one label, and for every pair $R \in \mathcal{R}$, $B \in \mathcal{B}$ of paths, if two vertices v and v' belong to both R and B, and are assigned the same label, then they appear in the same order on R and on B.*

Before we prove Theorem 2.2, let us first complete the proof of Theorem 2.1 assuming it. Let $\ell : V(H) \to L$ be the labeling computed by Theorem 2.2. Next, we switch S_1 and T_1, so that the directions of the paths in $\mathcal{R}$ are reversed. We apply Theorem 2.2 again to this new setting, and obtain another labeling $\ell' : V(H) \to L'$, where $L' = \{\ell'_1, \ell'_2, \ldots, \ell'_{2k}\}$.

Assume for contradiction that $|V(H)| \geq 4k^4 + 4k + 1$. Every non-terminal vertex v can be associated with a quadruple (R, B, ℓ_i, ℓ'_j), where R and B are the red and the blue paths on which v lies, ℓ_i is the label assigned to v by ℓ, and ℓ'_j is the label assigned to v by ℓ'. Since the total number of such quadruples is $4k^4$, there is a pair u, v of non-terminal vertices that have the same quadruple (R, B, ℓ_i, ℓ'_j). As u and v are assigned the same label by ℓ, they must appear in the same order on R and B. Assume w.l.o.g. that u appears before v on both these paths. However, since both these vertices are assigned the same label by ℓ', and since the red paths were reversed when computing ℓ', the order of u and v on paths R and B must be reversed, a contradiction. In order to complete the proof of Theorem 2.1, it now only remains to prove Theorem 2.2.

Proof of Theorem 2.2 Let $\tilde{H}$ be the directed counterpart of the graph H, where we direct all red edges along the direction of the red paths from S_1 to T_1, and we direct the blue edges similarly along the blue paths from S_2 to T_2. The main combinatorial object that we use in the proof is a chain. A chain Z is a directed (not necessarily simple) path in graph $\tilde{H}$, such that the edges of Z are alternating red and blue edges. In other words, if the edges of Z are $e_1, e_2, \ldots, e_r$ in this order, then all odd-indexed edges are red and all even-indexed edges are blue, or vice versa. The rest of the proof consists of three steps. First, we show that every chain must be a simple path, so no vertex may appear twice on a chain. If this is not the case, we will show that $\mathcal{R}$ is not a unique set of paths routing (S_1, T_1), or that $\mathcal{B}$ is not a unique set of paths routing (S_2, T_2), leading to a contradiction. In the second step, we construct a collection of $2k$ chains using a natural greedy algorithm: start from some source, and then follow alternatively red and blue edges, while possible. We will show that every vertex of H belongs to at least one chain (but may belong to more than one). We then associate a separate label with each chain, and assign all vertices that belong to a chain the same label. If a vertex belongs to several chains, then one of the corresponding labels is assigned arbitrarily. Finally, we prove that for every path $P \in \mathcal{R} \cup \mathcal{B}$ and every chain Z, if v and v' are two vertices that belong to both P and Z, then they must appear in the same order on P and on Z.

Before we proceed, we define two auxiliary structures: red and blue cycles. Let C be a directed simple cycle in the graph $\tilde{H}$ (so every vertex may appear at most once on C). We say that it is a *blue cycle* iff we can partition C into an even number of edge-disjoint consecutive segments $\sigma_1, \sigma_2, \ldots, \sigma_{2r}$, where $r > 0$; for all $1 \leq i \leq r$, σ_{2i} consists of a single red edge, and σ_{2i-1} is a non-empty path that only consists of blue edges. Every edge of C belongs to exactly one segment, and every consecutive pair of segments shares one vertex (if $r = 1$ then the two segments share two vertices — the endpoints of the segments). A red cycle is defined similarly, with the roles of the red and the blue segments reversed. We start by showing that $\tilde{H}$ cannot contain a red or a blue cycle. Proofs of several claims in the rest of the section are omitted and can be found in [10].

Lemma 2.1. *Graph $\tilde{H}$ cannot contain a red cycle or a blue cycle.*

The claim below essentially follows from the preceding lemma.

Claim 2.2. *If Z is a chain, then every vertex of $V(H)$ may appear on Z at most once.*

We define a collection $\mathcal{Z}$ of $2k$ chains in $\tilde{H}$, and prove that every vertex of $\tilde{H}$ belongs to at least one chain. Let $s \in S_1 \cup S_2$, and let e be the unique edge leaving s. We start building the chain by adding e to the chain. If the last edge added to the chain $e' = (u, v)$ is a red edge, and there is a blue edge leaving v in $\tilde{H}$, then we add the unique blue edge leaving v in $\tilde{H}$ to the chain; if no such edge exists, we complete the construction of the chain — in this case, $v \in T_1 \cup T_2$ must hold. Similarly, if the last edge added to the chain $e' = (u, v)$ is a blue edge, and there is a red edge leaving v in $\tilde{H}$, then we add the unique red edge leaving v in $\tilde{H}$ to the chain; if no such edge exists, we complete the construction of the chain. Overall, we construct one chain starting from each vertex in $S_1 \cup S_2$, obtaining $2k$ chains. Let $\mathcal{Z}$ denote the resulting collection of the chains.

Claim 2.3. *Every vertex of $\tilde{H}$ belongs to at least one chain.*

Our final step is the following claim.

Claim 2.4. *Let Z be a chain, and assume that it contains two vertices $v, v' \in V(P)$, where $P \in \mathcal{R} \cup \mathcal{B}$. Assume further that v appears before v' on Z. Then v appears before v' on P.*

We are now ready to assign labels to the vertices of H. Let $\mathcal{Z} = \{C_1, C_2, \ldots, C_{2k}\}$. Fix any vertex $v \in H$, and let $C_i \in \mathcal{Z}$ be any chain that contains v. We then assign to v the label ℓ_i.

Consider now any pair $R \in \mathcal{R}$, $B \in \mathcal{B}$ of paths, and let v, v' be two vertices that have the same label ℓ_i and appear on both R and B. Assume w.l.o.g. that v appears before v' on chain C_i. Then from Claim 2.4, v must appear before v' on both R and B. This completes the proof of Theorem 2.2, and hence of Theorem 2.1 and Theorem 1.2.

3 Background on Treewidth and Path-of-Sets System

In this section we define some graph-theoretic notions and summarize some previous results that we use in the proof of Theorem 1.1. We also define a combinatorial object that plays a central role in the proof — the path-of-sets system from [9].

Given a graph $G = (V, E)$ and a set $A \subseteq V$ of vertices, we denote by $E_G(A)$ the set of edges with both endpoints in A, and by $\text{out}_G(A)$ the set of edges with exactly one endpoint in A. For disjoint sets of vertices A and B, the set of edges with one end point in A and the other in B is denoted by $E_G(A, B)$. For a vertex v in a graph G we use $d_G(v)$ to denote its degree. We may omit the subscript G if it is clear from the context. Given a set $\mathcal{P}$ of paths in G, we denote by $V(\mathcal{P})$ the set of all vertices participating in paths in $\mathcal{P}$, and similarly, $E(\mathcal{P})$ is the set of all edges that participate in paths in $\mathcal{P}$. We sometimes refer to sets of vertices as *clusters*. A path P in a graph G is a 2*-path* iff every inner vertex v in P has $d_G(v) = 2$. It is a *maximal* 2*-path* iff the degrees of the endpoints of P are both different from 2. Given a set $\mathcal{P}$ of paths, we denote by $J(\mathcal{P})$ the graph obtained by the union of all the paths in $\mathcal{P}$. Given a graph H, let $\tau(H)$ denote the number of vertices of H whose degree is more than 2 in H.

We now define the notion of linkedness and the different notions of well-linkedness that we use.

Definition 1. We say that a set $\mathcal{T}$ of vertices is α-well-linked[3] in G, iff for any partition (A, B) of the vertices of G into two subsets, $|E(A, B)| \geq \alpha \cdot \min\{|A \cap \mathcal{T}|, |B \cap \mathcal{T}|\}$.

Definition 2. We say that a set $\mathcal{T}$ of vertices is *node-well-linked* in G, iff for any pair $(\mathcal{T}_1, \mathcal{T}_2)$ of equal-sized subsets of $\mathcal{T}$, there is a collection $\mathcal{P}$ of $|\mathcal{T}_1|$ **node-disjoint** paths, connecting the vertices of $\mathcal{T}_1$ to the vertices of $\mathcal{T}_2$. (Note that $\mathcal{T}_1$, $\mathcal{T}_2$ are not necessarily disjoint, and we allow empty paths).

The two different notions of well-linkedness are closely related. In particular, suppose $\mathcal{T}$ is α-well-linked in a graph G of maximum degree Δ. Then there is a large subset $\mathcal{T}' \subseteq \mathcal{T}$ of vertices that is node-well-linked in G, as shown in the following theorem.

Theorem 3.1. (Theorem 2.2 in [9]) *Suppose we are given a connected graph $G = (V, E)$ with maximum vertex degree Δ, and a subset $\mathcal{T}$ of κ vertices called*

[3] This notion of well-linkedness is based on edge-cuts and we distinguish it from node-well-linkedness that is directly related to treewidth. For technical reasons it is easier to work with edge-cuts and hence we use the term well-linked to mean edge-well-linkedness, and explicitly use the term node-well-linkedness when necessary.

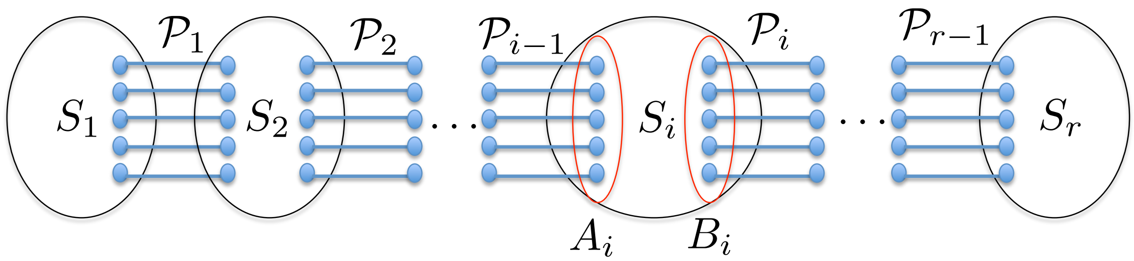

Figure 2: Path-of-Sets System

terminals, such that $\mathcal{T}$ is α-well-linked in G, for some $\alpha < 1$. Then there is a subset $\mathcal{T}' \subset \mathcal{T}$ of $\Omega\left(\frac{\alpha\kappa}{\Delta}\right)$ terminals, such that $\mathcal{T}'$ is node-well-linked in G.

The following well-known lemma summarizes an important connection between treewidth and node-well-linkedness.

Lemma 3.1. ([35]) *Let k be the size of the largest node-well-linked set in G. Then $k \leq \mathrm{tw}(G) \leq 4k$.*

Combining Theorem 3.1 with Lemma 3.1, we obtain the following theorem.

Theorem 3.2. *Let G be any graph with maximum vertex degree Δ, and $\mathcal{T}$ a subset of κ vertices, such that $\mathcal{T}$ is α-well-linked in G, for $\alpha < 1$. Then the treewidth of G is $\Omega(\alpha\kappa/\Delta)$.*

A notion closely related to well-linkedness is that of linkedness, where we require good connectivity between a pair of disjoint vertex subsets.

Definition 3. We say that two disjoint vertex subsets A and B are *linked* in G iff for any pair of equal-sized subsets $A' \subseteq A$, $B' \subseteq B$ there is a set $\mathcal{P}$ of $|A'|$ node-disjoint paths connecting A' to B' in G.

Path-of-Sets System A central combinatorial object that we use in the proof of Theorems 1.1 is a path-of-sets system, that was introduced in [9] (a somewhat similar object, called a grill, was introduced by Leaf and Seymour [27]). See Figure 2.

Definition 4. A path-of-sets system $(\mathcal{S}, \bigcup_{i=1}^{r-1} \mathcal{P}_i)$ of width r and height h consists of:

- A sequence $\mathcal{S} = (S_1, \ldots, S_r)$ of r disjoint vertex subsets of G, where for each i, $G[S_i]$ is connected;
- For each $1 \leq i \leq r$, two disjoint sets $A_i, B_i \subseteq S_i$ of h vertices each, such that A_i and B_i are linked in $G[S_i]$;
- For each $1 \leq i < r$, a set $\mathcal{P}_i$ of h disjoint paths, routing (B_i, A_{i+1}), such that all paths in $\bigcup_i \mathcal{P}_i$ are mutually disjoint, and do not contain the vertices of $\bigcup_{S_i \in \mathcal{S}} S_i$ as inner vertices,

We say that it is a *strong* path-of-sets system, if additionally for each $1 \leq i \leq r$, A_i is node-well-linked in $G[S_i]$, and the same holds for B_i.

The following theorem, that was proved in [9], is the starting point of the proof of Theorem 1.1.

Theorem 3.3. (Theorem 3.2 in [9]) *Let G be any graph of treewidth k, and let $h, r > 1$ be integral parameters, such that for some large enough constants c and c', $k/\log^{c'} k > chr^{48}$. Then there is an efficient randomized algorithm, that, given G, h and r, w.h.p. computes a strong path-of-sets system of height h and width r in G.*

Expanders and the Cut-Matching Game. We say that a (multi)-graph $G = (V, E)$ is an α-expander, iff $\min_{\substack{S \subseteq V:\\ |S| \leq |V|/2}} \left\{ \frac{|E(S, \overline{S})|}{|S|} \right\} \geq \alpha$. We use the cut-matching game of Khandekar, Rao and Vazirani [24] to construct an expander that can be appropriately embedded in a graph. In this game, we are given a set V of N vertices, where N is even, and two players: a cut player, whose goal is to construct an expander X on the set V of vertices, and a matching player, whose goal is to delay its construction. The game is played in iterations. We start with the graph X containing the set V of vertices, and no edges. In each iteration j, the cut player computes a bi-partition (A_j, B_j) of V into two equal-sized sets, and the matching player returns some perfect matching M_j between the two sets. The edges of

M_j are then added to X. Khandekar, Rao and Vazirani have shown that there is a strategy for the cut player, guaranteeing that after $O(\log^2 N)$ iterations, no matter the strategy of the matching player, the resulting graph is a $\frac{1}{2}$-expander w.h.p. Subsequently, Orecchia et al. [33] have shown the following improved bound:

THEOREM 3.4. ([33]) *There is a probabilistic algorithm for the cut player, such that, no matter how the matching player plays, after $\gamma_{\text{CMG}}(N) = O(\log^2 N)$ iterations, graph X is an $\alpha_{\text{CMG}}(N) = \Omega(\log N)$-expander, with constant probability.*

Our algorithms work by embedding an expander X into a sub-graph of G. The embedding of the expander is then used to certify the treewidth. We use the following notion of embedding.

Definition 5. Let G, X be graphs. An embedding φ of X into G maps every vertex $v \in X$ to a connected subgraph $C_v \subseteq G$, and every edge $e = (u, v) \in E(X)$ to a path P_e in graph G, whose endpoints belong to C_v and C_u, respectively. We say that the *congestion* of the embedding is at most c, iff every edge of G belongs to at most $c-1$ paths in $\{P_e \mid e \in E(X)\}$ and at most one graph $\{C_v \mid v \in V(X)\}$.

In the next simple claim, we show that if we can embed a κ-vertex expander with congestion at most c into a graph H with bounded vertex degree, then the treewidth of H is large.

CLAIM 3.2. *Let X be an α-expander on κ vertices for $\alpha < 1$, with maximum vertex degree Δ', and let H be a graph with maximum vertex degree at most Δ, such that that there is an embedding of X into H with congestion η. Then* $\text{tw}(H) = \Omega\left(\frac{\alpha\kappa}{\eta\Delta\Delta'}\right)$.

4 A Small Treewidth-Preserving Degree-4 Minor

In this section we prove the following theorem which gives a degree-4 sparsifier.

THEOREM 4.1. *There is a randomized algorithm, that, given a graph G of treewidth at least k, w.h.p. computes a minor H of G, such that:*

- *the treewidth of H is $\Omega(k/\text{poly}\log k)$;*
- *every vertex has degree at most 4 in H; and*
- $|V(H)| = O(k^4)$.

The running time of the algorithm is polynomial in $|V(G)|$ and k.

In order to prove Theorem 4.1, it is sufficient to find a subgraph H of G, with $\tau(H) = O(k^4)$, such that the maximum vertex degree in H is at most 4, and the treewidth of H is $\Omega(k/\text{poly}\log k)$. Indeed, by replacing every maximal 2-path in H with an edge connecting its endpoints, we obtain the desired minor.

We start by applying Theorem 3.3 with $r = \gamma_{\text{CMG}}(k)$ and $h = \Omega(k/\text{poly}\log k)$, so that h is an even integer, and $\frac{k}{\log^{c'} k} > chr^{48}$ holds, where c, c' are the constants from Theorem 3.3. Let $(\mathcal{S}, \bigcup_{i=1}^{r-1} \mathcal{P}_i)$ be the resulting path-of-sets system.

Our next step is to construct an expander graph X on h vertices, and to embed it into a sub-graph H of G. Following the previous work on routing problems [34, 1, 14, 15, 11], we will embed X into G using the cut-matching game, and the path-of-sets system $(\mathcal{S}, \bigcup_{i=1}^{r-1} \mathcal{P}_i)$. We start with an intuitive high-level description of the algorithm. For each $1 \le i \le r$, let $\mathcal{Q}_i$ be any set of node-disjoint paths connecting A_i to B_i in $G[S_i]$ (this set exists due to the linkedness of (A_i, B_i) in $G[S_i]$). Let $\mathcal{H}$ be the set of h paths, obtained by concatenating $\mathcal{Q}_1, \mathcal{P}_1, \mathcal{Q}_2, \mathcal{P}_2, \ldots, \mathcal{P}_{r-1}, \mathcal{Q}_r$. We denote $\mathcal{H} = \{P_1, \ldots, P_h\}$. The high-level idea is to construct an expander X over a set $V = \{v_1, \ldots, v_h\}$ of h vertices, and to embed it into G using the cut-matching game, as follows. For each $1 \le i \le h$, we embed v_i into P_i, that is, $C_{v_i} = P_i$. We construct the edges of the expander, and embed them into G, using the cut-matching game, where for each $1 \le j \le r$, we use cluster $S_j \in \mathcal{S}$ to route the jth matching, as follows. A partition (Y, Z) of the vertices of X computed by the cut player naturally defines a partition $(\mathcal{H}_Y, \mathcal{H}_Z)$ of the paths in $\mathcal{H}$ into two equal-sized subsets, which in turn defines a partition (A'_j, A''_j) of A_j into two equal-sized subset. Using the fact that A_j is node-well-linked inside $G[S_j]$, we can find a set $\mathcal{B}_j$ of node-disjoint paths in $G[S_j]$ connecting A'_j to A''_j. This set of paths defines a matching M_j between the paths in $\mathcal{H}_Y$ and $\mathcal{H}_Z$, and hence between the vertices of Y and Z in X. We view this matching as the response of the matching player. After $\gamma_{\text{CMG}}(h) \le \gamma_{\text{CMG}}(k) = r$ iterations, we obtain an expander X and its embedding with congestion 2 into G. Intuitively, we would like to define H as the set of all edges and vertices of G used in this embedding, that is, the union of the paths in $\mathcal{H}$ and $\bigcup_{j=1}^{r} \mathcal{B}_j$. It is easy to see that the maximum vertex degree in H is at most 4, since in each cluster S_j we only route 2 sets of node-disjoint paths: $\mathcal{Q}_j$ and $\mathcal{B}_j$. However, $\tau(H)$ may not be bounded by $O(k^4)$. In particular, the paths in $\mathcal{Q}_j$ and $\mathcal{B}_j$ may intersect at many vertices. In order to overcome this difficulty, we can use Theorem 1.2 to find new sets $\mathcal{Q}'_j$ and $\mathcal{B}'_j$ of paths, routing the same pairs of vertex subsets, such that, if $J = J(\mathcal{Q}'_j \cup \mathcal{B}'_j)$, then $\tau(J) = O(h^4)$. However, this re-routing changes the paths in $\mathcal{H}$, and therefore the mapping between the vertices in sets $A_{j'}$ for $j' \neq j$ and the vertices in X may be changed. Therefore, we need to execute this procedure more carefully. In particular, we apply Theorem 1.2 in the graph $G[S_j]$ after each iteration j of the cut-matching game; for iteration $j+1$ we exploit the node-well-linkedness of the set A_{j+1} in $G[S_{j+1}]$ to maintain consistency in the mapping of paths in $\mathcal{H}$ to the vertices of the expander. The formal description of the embedding procedure can be found in [10].

5 Building a Degree-3 Minor

In this section we complete the proof of Theorem 1.1. We start with an informal overview to help understand the high-level plan.

5.1 Overview We use an algorithm, similar to the one used in Section 4, in order to embed an expander into G, using the path-of-sets system. The main difference is that, instead of embedding a single expander X, we will embed N expanders $X_1, \ldots, X_N$ where $N = \Theta(\log k)$. For this purpose we start with a longer path-of-sets system $(\mathcal{S}, \bigcup_{i=1}^{r-1} \mathcal{P}_i)$ with parameters $h = k/\operatorname{poly}\log k$ and $r = O(\log^3 k)$ and partition it into $N = O(\log k)$ smaller path-of-sets systems with parameters $r^* = \gamma_{\text{CMG}}(h)$ and h (hence $r \simeq Nr^*$). For $1 \leq i \leq N$, we embed an expander X_i into the i'th path-of-sets system using the approach in the preceding section. Recall that for each cluster S_i, we construct two sets of paths contained in $G[S_i]$: one set, $\mathcal{R}_i$, that we call red paths, routes (A_i, B_i), and another set, $\mathcal{B}_i$, that we call blue paths, routes (A'_i, A''_i), where (A'_i, A''_i) is the partition of A_i defined by the cut player. Let H_i be the topological minor of $G[S_i]$ obtained by taking the union of the paths in $\mathcal{R}_i$ and $\mathcal{B}_i$, and suppressing all degree-2 vertices, except for $A_i \cup B_i$. We assume that H_i is minimal in the following sense: for each edge e of H_i, either (A_i, B_i) or (A'_i, A''_i) is not routable in $H_i \setminus \{e\}$. Abusing the notation, we assume that $\mathcal{R}_i$ and $\mathcal{B}_i$ are the sets of the red and the blue paths, routing (A_i, B_i) and (A'_i, A''_i), respectively in H_i. Notice that every vertex of H_i must lie on some red path. Let $\mathcal{H}$ be the set of h paths obtained by concatenating the paths in $\mathcal{R}_1, \mathcal{P}_1, \ldots, \mathcal{P}_{Nr^*-1}, \mathcal{R}_{Nr^*}$, and let H be the topological minor of G obtained by taking the union of the graphs H_i and the paths $\bigcup_{i=1}^{Nr^*-1} \mathcal{P}_i$. We say that an edge of H is a red edge if it belongs to a red path and no blue paths; it is a blue edge if it belongs to a blue path and no red paths; and it is a red-blue edge if it belongs to both a red and a blue path. We can view the N different expanders as sharing the same vertex set, where the vertices correspond to the paths in $\mathcal{H}$. Consider a vertex v of degree 4 in graph H; it must be incident to two red edges and two blue edges. In order to reduce the degree to 3, we use random sampling to pick one of the two blue edges incident to v and eliminate it. After this step the degree of every vertex is at most 3. Let H^* be this final topological minor of G. The heart of the analysis is to show that H^* has treewidth $\Omega(k/\operatorname{poly}\log k)$. This is done by showing that the set $A = A_1$ of vertices remains α-well-linked in H^*, for $\alpha = \Omega(1/\operatorname{poly}\log k)$, and applying Theorem 3.2.

We start by observing that the set A of vertices is α_{WL}-well-linked in H, for some constant α_{WL}. This is shown by using the embeddings of the expanders $X_1, \ldots, X_N$ into H. Next, we carefully partition each path in $\mathcal{H}$ into a collection of disjoint segments. Intuitively, each segment of a path $P \in \mathcal{H}$ is a sub-path of P of length $\Theta(\operatorname{poly}\log h)$. We then contract each such segment σ into a super-node v_σ. Let F be this contracted graph, and let F^* be the corresponding contracted graph of H^*. Equivalently, F^* is obtained from F by deleting all the edges in $E(H) \setminus E(H^*)$.

Each vertex of A belongs to a distinct contracted segment, and is associated with the corresponding super-node in F. We do not distinguish between the vertices of A and their corresponding super-nodes. It is easy to see that A remains α_{WL}-well linked in F since we only contracted edges. The most crucial property of the contracted graph F is that the value of the minimum cut in F is at least $\Omega(\log |V(F)|)$. This allows us to use arguments similar to those used in Karger's sampling technique [22] to show that all cuts are approximately preserved in F^*. In particular, the vertices of A remain $\alpha_{\text{WL}}/32$-well-linked in F^*. Since the length of every segment used in the construction of the contracted graph F is $O(\operatorname{poly}\log h)$, this implies that the vertices of A are α-well-linked in H^*, for $\alpha = \Omega(1/\operatorname{poly}\log k)$. The most challenging part of the proof is to set up the partition of the paths in $\mathcal{H}$ into segments, so that in the resulting contracted graph F, the value of the minimum cut is $\Omega(\log |V(F)|)$. At a high-level, the proof proceeds as follows. Assume for contradiction, that there is a partition (X, Y) of $V(F)$ with $X, Y \neq \emptyset$, and $|E_F(X, Y)| < N$. Let $X' \subseteq V(H)$ be obtained from X by un-contracting all super-nodes in X, and let $Y' \subseteq V(H)$ be obtained from Y similarly. Then (X', Y') is a partition of $V(H)$, and $|E_H(X', Y')| < N$. Assume first that there are two paths $P, P' \in \mathcal{H}$, such that $P \subseteq H[X']$ and $P' \subseteq H[Y']$. We then use the embeddings of the expanders $X_1, \ldots, X_N$ to argue that $|E_H(X', Y')| \geq N$, reaching a contradiction. Therefore, we can assume w.l.o.g. that no path of $\mathcal{H}$ is contained in $H[X']$. We next show that for some $1 \leq i^* \leq Nr^*$, partition (X', Y') of $V(H)$ defines a partition (X^*, Y^*) of $V(H_{i^*})$, such that $|X^*|, |Y^*| > 200N^4$, while $|E_{H_{i^*}}(X^*, Y^*)| < N$. We then consider the segments of the red paths in $\mathcal{R}_{i^*}$ and the blue paths in $\mathcal{B}_{i^*}$ that are contained in $H_{i^*}[X^*]$. Let $\mathcal{R}^*$ denote the corresponding segments of the red paths, and $\mathcal{B}^*$ the corresponding segments of the blue paths. Using Theorem 2.1, we show that there is some edge $e \in H[X^*]$, such that we can still route the endpoints of the paths in $\mathcal{R}^*$ to each other, and the endpoints of the paths in $\mathcal{B}^*$ to each other, even after deleting e from $H[X^*]$. This new routing implies that we can route both (A_{i^*}, B_{i^*}) and (A'_{i^*}, A''_{i^*}) in $H_{i^*} \setminus \{e\}$, contradicting the minimality of H_{i^*}.

5.2 Proof of Theorem 1.1 We set $r = 2^{15} \log k \cdot \gamma_{\text{CMG}}(k) = \Theta(\log^3 k)$, and $h = \Omega(k/\operatorname{poly}\log k)$, so that h is an even integer, and $k/\log^{c'} k > chr^{48}$, where c and c' are the constants from Theorem 3.3. We assume w.l.o.g. that k is large enough, so $h > 72 \log k$ and $h > \gamma_{\text{CMG}}(h)$. We then apply Theorem 3.3 to graph G, with parameters r and h, to obtain a strong path-of-sets system $(\mathcal{S}, \bigcup_{i=1}^{r-1} \mathcal{P}_i)$ of height h and width r.

Let $r^* = \gamma_{\text{CMG}}(h)$, and let $N = \lceil 3072 \log(10h^4 \cdot r^*) \rceil$; it is easy to see that $N = \Theta(\log h)$. We will assume w.l.o.g. that h is large enough, so $N > 1536 \log(10h^4 \cdot r^* \cdot N)$ holds. Finally, we let $r' = N \cdot r^*$. Note that $r' = r^* \cdot \lceil 3072 \log(10h^4 \cdot r^*) \rceil \leq 2^{15} \gamma_{\text{CMG}}(h) \log h < r$.

We construct a new, smaller, path-of-sets system, of height h and width r', using the clusters $\mathcal{S}' = (S_1, \ldots, S_{r'})$, and the sets $\mathcal{P}_i$ of paths, for $1 \leq i \leq r'-1$; in other words we restrict attention to the first r' clusters from the initial path-of-sets system. Abusing notation, we denote r' by r and $\mathcal{S}'$ by $\mathcal{S}$.

We denote by G' the following minor of G: start with the union of $G[S_i]$ for all $1 \leq i \leq r$; for each path $P \in \bigcup_{i=1}^{r-1} \mathcal{P}_i$, add an edge connecting the endpoints of P to G'. We denote by E_i the set of edges corresponding to the paths in $\mathcal{P}_i$. Equivalently, we obtain G' from graph $\left(\bigcup_{S_i \in \mathcal{S}} G[S_i]\right) \cup \left(\bigcup_{j=1}^{r-1} \mathcal{P}_j\right)$ by suppressing degree-2 internal nodes on the paths in $\bigcup_{i=1}^{r-1} \mathcal{P}_i$. It is now enough to find a topological minor H^* of G' whose treewidth is $\Omega(k/\operatorname{poly}\log k)$, maximum vertex degree is 3, and $|V(H^*)| = O(k^4)$. We do so via the following theorem:

THEOREM 5.1. *There is an efficient randomized algorithm, that finds a topological minor H^* of G', such that w.h.p.:*

- $|V(H^*)| = O(h^4 \cdot r)$;
- *The maximum vertex degree in H^* is* 3;
- $A_1 \subseteq V(H^*)$; *and*
- *The set A_1 of vertices is α-well-linked in H^*, for $\alpha = \Omega(1/\log^7 k)$.*

Theorem 1.1 follows easily from Theorem 5.1. The desired topological minor of G is H^*. The only property that is left to verify is that $\operatorname{tw}(H) = \Omega(k/\text{polylog(k)})$ which follows from α-well-linkedness of A_1 in H^*. Indeed, Theorem 3.2 implies that $\operatorname{tw}(H) = \Omega(\alpha|A_1|/3) = \Omega(k/\text{polylog(k)})$ since $|A_1| = h = \Omega(k/\text{polylog(k)})$, $\alpha = \Omega(1/\log^7 k)$ and H^* has maximum degree 3. From now on we focus on proving Theorem 5.1.

In order to simplify the notation, we refer to the graph G' as G. Recall that we are given a path-of-sets system $(\mathcal{S} = (S_1, \ldots, S_r), \bigcup_{i=1}^{r-1} \mathcal{P}_i)$ of height h and width $r = Nr^*$ in G, where for each $1 \leq i < r$, each path in $\mathcal{P}_i$ consists of a single edge, and the corresponding set of edges is denoted by E_i. Let $E' = \bigcup_{i=1}^{r-1} E_i$. We denote A_1 by A. Our goal is to construct a topological minor H^* of G, such $|V(H^*)| = O(h^4 r)$, the maximum vertex degree of H^* is 3, while ensuring that $A \subseteq V(H^*)$ and it is α-well-linked in H^*, w.h.p.

The rest of the proof consists of three steps. In the first step, we define the sets $\mathcal{B}_i, \mathcal{R}_i$ of paths for $1 \leq i \leq r$ by playing the cut-matching games; in the second step we partition the resulting red paths into segments; and in the third step we complete the proof of the theorem.

Step 1: Cut-Matching Games In this step we construct N expanders $X_1, \ldots, X_N$, and embed each of them separately into G. For each $1 \leq i \leq N$, let $\mathcal{S}_i = (S_{(i-1)r^*+1}, \ldots, S_{ir^*})$, let $E^i = \bigcup_{j=(i-1)r^*+1}^{ir^*-1} E_j$, and let $\hat{E}^i = E_{ir^*}$ (for $i = N$, $\hat{E}^i = \emptyset$). Let G_i be the graph obtained from the union of $G[S_j]$ for all $S_j \in \mathcal{S}_i$ and the edges in E^i. For each $1 \leq i \leq N$, we embed the expander X_i into G_i, using the cut-matching game, as follows. For convenience, we denote $(i-1)r^*$ by z.

We will gradually construct a set $\mathcal{H}_i$ of paths over the course of r^* iterations. For each $1 \leq j \leq r^*$, at the beginning of the jth iteration, we are given a set $\mathcal{H}^j$ of h disjoint paths, connecting the vertices of A_{z+1} to the vertices of A_{z+j}, and a bijection $f : \mathcal{H}^j \to V(X_i)$. At the beginning, $\mathcal{H}^1$ consists of h paths, each of which consists of a single distinct vertex of A_{z+1}, and the mapping $f : \mathcal{H}^1 \to V(X_i)$ is an arbitrary bijection. We also start with a graph X_i on h vertices, and $E(X_i) = \emptyset$. For $1 \leq j \leq r^*$, the jth iteration is executed as follows.

We use the cut player on the current graph X_i to find a partition (Y_j, Z_j) of $V(X_i)$ into two equal-sized subsets. This naturally defines a partition $(\mathcal{H}^j_Y, \mathcal{H}^j_Z)$ of $\mathcal{H}^j$ where $\mathcal{H}^j_Y$ contains all paths $P \in \mathcal{H}^j$, such that $f(P) \in Y_j$. In turn, this gives a partition (A'_{z+j}, A''_{z+j}) of A_{z+j}, where a vertex $v \in A_{z+j}$ belongs to A'_{z+j} iff the path P on which v lies belongs to $\mathcal{H}^j_Y$. Since the set A_{z+j} of vertices is node-well-linked in $G[S_{z+j}]$, there is a collection of node-disjoint paths routing (A'_{z+j}, A''_{z+j}) in $G[S_{z+j}]$. Since A_{z+j} and B_{z+j} are linked in $G[S_{z+j}]$, there is a collection of node-disjoint paths routing (A_{z+j}, B_{z+j}) in $G[S_{z+j}]$. From Theorem 1.2, we can find a set $\mathcal{B}'_{z+j}$ of paths routing (A'_{z+j}, A''_{z+j}) , and a set $\mathcal{R}'_{z+j}$ of paths routing (A_{z+j}, B_{z+j}) in $G[S_{z+j}]$, such that, if $J = J(\mathcal{B}'_{z+j} \cup \mathcal{R}'_{z+j})$, then the maximum vertex degree in J is bounded by 4, the degree of every vertex in $A_{z+j} \cup B_{z+j}$ is at most 3, and $\tau(J) \leq 8h^4 + 8h$. We will assume that J is a minimal graph in which (A'_{z+j}, A''_{z+j}) and (A_{z+j}, B_{z+j}) are both routable: that is, for every edge $e \in E(J)$, either (A'_{z+j}, A''_{z+j}), or (A_{z+j}, B_{z+j}) are not routable in $J \setminus \{e\}$. We let H_{z+j} be the graph obtained from J by replacing every maximal 2-path that does not contain the vertices of $A_{z+j} \cup B_{z+j}$ as inner vertices, by an edge connecting its two endpoints. Then $|V(H_{z+j})| \leq 8h^4 + 8h \leq 10h^4$, every vertex of H_{z+j} has degree at most 4, while the vertices in $A_{z+j} \cup B_{z+j}$ have degree at most 3; there is a set $\mathcal{B}_{z+j}$ of paths routing (A'_{z+j}, A''_{z+j}) , and a set $\mathcal{R}_{z+j}$ of paths routing (A_{z+j}, B_{z+j}) in H_{z+j}, and for every edge $e \in E(H_{z+j})$, either (A'_{z+j}, A''_{z+j}), or (A_{z+j}, B_{z+j}) are not routable in $H_{z+j} \setminus \{e\}$. We call the paths in $\mathcal{R}_{z+j}$ *red paths*, and the paths in $\mathcal{B}_{z+j}$ *blue paths*. An edge that belongs to a red path, but no blue paths is called a red edge. An edge the belongs to a blue path but no red paths is called a blue edge. An edge that lies on a red and a blue path is called a red-blue edge. Notice that a vertex of H_{z+j} has degree 4 only if it is incident on two blue edges. Each vertex in A_{z+j} serves as a source of a red path and a source or a destination of a blue path, so it can only be incident on at most two edges in H_{z+j}. A vertex $v \in B_{z+j}$ serves as a destination of a red path; its degree is at most 3, and it is equal to 3 only if v is incident on two blue edges.

We let $\mathcal{H}^{j+1}$ be the concatenation of the paths in $\mathcal{H}^j$, $\mathcal{R}_{z+j}$, and E_{z+j}. In order to define the mapping $f : \mathcal{H}^{j+1} \to V(X_i)$, for each $P \in \mathcal{H}^{j+1}$, let $P' \in \mathcal{H}^j$ be the sub-path of P. Then we set $f(P) = f(P')$. Notice that the set $\mathcal{B}_{z+j}$ of paths defines a matching between the paths in $\mathcal{H}^j_Y$ and $\mathcal{H}^j_Z$, which in turn naturally defines a matching M_j between Y_j and Z_j in X_i. We add the edges of the matching M_j to X. Each edge $e = (v_i, v_{i'}) \in M_j$ is mapped to the corresponding path in $\mathcal{B}_{z+j}$, that connects the unique vertex in $A_j \cap f^{-1}(v_i)$ to the unique vertex in $A_j \cap f^{-1}(v_{i'})$.

Finally, we set $\mathcal{H}_i = \mathcal{H}^{r^*}$. Let $\tilde{H}_i$ be the union of the graphs $H_{z+1}, \ldots, H_{z+r^*}$, and the edges E^i. Then we have defined an $\alpha_{\text{CMG}}(h)$-expander X_i on h vertices with maximum vertex degree $\gamma_{\text{CMG}}(h)$, and embedded it with congestion 2 into $\tilde{H}_i$, where each vertex of X_i is embedded into a distinct path in $\mathcal{H}_i$.

Let H be the union of the graphs $\tilde{H}_i$, for $1 \leq i \leq N$ and $\bigcup_{i=1}^{N-1} \hat{E}^i$, and let $\mathcal{H}$ be the concatenation of $\mathcal{H}_1, \hat{E}_1, \ldots, \hat{E}_{N-1}, \mathcal{H}_N$. We will sometimes refer to the paths in $\mathcal{H}$ as red paths. All vertices in H have degree at most 4, and, as observed before, a vertex of H may have degree 4 only if it is incident on exactly two blue edges. Every vertex in A has degree at most 2. Our final graph H^* is obtained from H as follows: for each vertex $v \in (H)$ that is incident on two blue edges, we independently choose one of these two blue edges at random. Each blue edge that has been chosen by at least one vertex is then deleted from the graph. This final graph is denoted by H^*. Notice that each edge $e = (u, v)$ may be deleted from H due to the choice made by u, or the choice made by v; the overall probability that e is not deleted is at least $1/4$. Moreover, if e and e' do not share endpoints, then the events that e is deleted and that e' is deleted are independent.

It is immediate to see that $|V(H^*)| \leq Nr^* \cdot O(h^4) = O(rh^4)$; the vertices of A are contained in $V(H^*)$, and the maximum vertex degree in H^* is 3. It now only remains to prove that w.h.p. the vertices of A are α-well-linked in H^*, for some $\alpha = \Omega(1/\log^7 k)$. We do so in the next two steps, using the following claim whose proof can be found in [10].

Claim 5.1. *The set A of vertices is α_{WL}-well-linked in H, where* $\alpha_{\text{WL}} = \min\left\{\frac{1}{2}, \frac{N \cdot \alpha_{\text{CMG}}(h)}{4\gamma_{\text{CMG}}(h)}\right\} = \Omega(1)$.

Step 2: Partitioning the Red Paths In this step, we will define a collection Σ_P of disjoint segments for every path $P \in \mathcal{H}$.

Consider any such path $P \in \mathcal{H}$. A sub-path P' of P is called a *heavy sub-path* iff for some $1 \leq i \leq Nr^*$, P' contains at least $200N^4 = \Theta(\log^4 h)$ vertices that belong to H_i.

If P contains no heavy sub-paths, then $\Sigma_P = \{P\}$. Notice that P contains at most $Nr^* \cdot O(\log^4 h) = O(\log^7 h)$ vertices in this case. Otherwise, we perform a number of iterations. In each iteration, we start with some heavy sub-path P' of P, where at the beginning of the first iteration, $P' = P$. Let P'' be the minimum-length heavy sub-path of P' containing the first vertex of P'. If $P' \setminus P''$ is a heavy path, then we add P'' to Σ_P, delete all vertices of P'' from P', and continue to the next iteration. Otherwise, we add P' to Σ_P and finish the algorithm. Notice that in any case, the length of every path added to Σ_P is at most $Nr^* \cdot O(\log^4 h) = O(\log^7 h)$. Overall, for each path $P \in \mathcal{H}$, we obtain a partition of P into disjoint sub-paths of length at most $O(\log^7 h)$ each. Moreover, if $|\Sigma_P| > 1$, then each path in Σ_P is a heavy sub-path of P. Let $\Sigma = \bigcup_{P \in \mathcal{H}} \Sigma_P$.

We obtain a contracted graph F from H by contracting, for each $\sigma \in \Sigma$, the vertices of σ into a single super-node v_σ. For every vertex $u \in A$, let $g(u)$ be the super-node v_σ such that $u \in V(\sigma)$. Notice that for $u \neq u'$, $g(u) \neq g(u')$. Let $U = \{g(u) \mid u \in A\}$. Since, from Claim 5.1, the vertices of A are α_{WL}-well-linked in H, the vertices of U are α_{WL}-well-linked in F. Since every vertex of H must belong to some red path, $V(F) = \{v_\sigma \mid \sigma \in \Sigma\}$.

We define a graph F^* from H^*, by similarly contracting all segments in $\bigcup_{P \in \mathcal{H}} \Sigma_P$ into super-nodes. Equivalently, graph F^* is obtained from F, by deleting all edges in $E(H) \setminus E(H^*)$. We prove the following claim.

Claim 5.2. *Set U is $\alpha_{\text{WL}}/32$-well-linked in F^* w.h.p.*

From the above claim we can easily show that A is α-well-linked in H^* since each node in H^* corresponds to the contraction of $O(\text{poly} \log h)$ nodes in the graph F^*; see [10] for details.

Step 3: Finishing the Proof In this step we prove Claim 5.2. We will sometimes refer to a subset $S \subseteq V(F)$ of the vertices of F, with $S, V(F) \setminus S \neq \emptyset$, as a *cut*. The value of the cut S is $|\text{out}(S)|$. The crucial part of the proof is the following claim whose proof can be found in [10].

Claim 5.3. *The value of the minimum cut in graph F is at least N.*

Let $n' = |V(H)|$. Then $|V(F)| \leq n' \leq 10h^4 \cdot r^* \cdot N$, and since $N > 1536 \log(10h^4 \cdot r^* \cdot N) \geq 1536 \log n'$, the value of the minimum cut in F is at least $1536 \log n'$. The number of edges in F is bounded by $m \leq 4n' \leq 40h^4 r^* N = O(h^4 \log^3 h)$. We use the following theorem of Karger:

Theorem 5.2. (Corollary A.6 in [22]) *Let G be any n-vertex graph, and assume that the value of the minimum cut in G is C. Then for any half-integral β, the number of cuts of value at most βC in G is bounded by $n^{2\beta}$.*

Since in graph F, the set U of vertices is α_{WL}-well-linked, it is enough to show that w.h.p., for any subset S of vertices of F, $|\text{out}_{F^*}(S)| \geq |\text{out}_F(S)|/32$. We partition the cuts $S \subseteq V(F)$ into $\lceil \log m \rceil$ collections $\mathcal{C}_1, \ldots, \mathcal{C}_{\lceil \log m \rceil}$, where for each $1 \leq i \leq \lceil \log m \rceil$, $\mathcal{C}_i$ contains all cuts S with $2^{i-1}N < |\text{out}_F(S)| \leq 2^i N$; set $\mathcal{C}_1$ also contains all cuts S with $|\text{out}_F(S)| = N$. Consider now some such collection $\mathcal{C}_i$. From Theorem 5.2, $|\mathcal{C}_i| \leq (n')^{2^{i+1}}$. Consider some set $S \in \mathcal{C}_i$. Let $S' \subseteq V(H)$ be obtained by un-contracting all super-nodes in S, that is, $S' = \bigcup_{v_\sigma \in S} V(\sigma)$. Notice that $\text{out}_H(S') = \text{out}_F(S)$, and $\text{out}_{H^*}(S') = \text{out}_{F^*}(S)$. Let $E_1(S) \subseteq \text{out}_H(S')$ contain all red and red-blue edges of $\text{out}_H(S')$, and let $E_2(S) = \text{out}_H(S') \setminus E_1(S)$. If $|E_1(S)| \geq |\text{out}_H(S')|/8$, then, since all edges of $E_1(S)$ belong to F^*, $|\text{out}_{F^*}(S)| \geq |\text{out}_F(S)|/8$. We assume from now on that this is not the case, and so $|E_2(S)| \geq 7|\text{out}_F(S)|/8$. Next, we construct a maximal set $E' \subseteq E_2(S)$ of edges, such that the edges in E' do not share endpoints in graph H. This is done by a simple greedy algorithm: while $E_2(S) \neq \emptyset$, let $e \in E_2(S)$ be any edge. Add e to E', and delete from $E_2(S)$ edge e and all edges sharing endpoints with e in graph H. Since all edges in $E_2(S)$ are blue, and each vertex may be incident on at most two blue edges, for every edge added to E', we delete at most three edges from $E_2(S)$. Therefore, eventually $|E'| \geq |E_2(S)|/3 \geq 7|\text{out}_H(S')|/24 \geq |\text{out}_H(S')|/4 = |\text{out}_F(S)|/4$ holds.

Each edge of E' belongs to $\text{out}_{F^*}(S)$ independently with probability at least 1/4. The expected number of the edges of E' that belong to $\text{out}_{F^*}(S)$ is therefore at least $|E'|/4 \geq |\text{out}_F(S)|/16 \geq N \cdot 2^{i-5}$.

We use the following standard Chernoff bound: let $X_1, \ldots, X_n$ be independent random variables in $\{0,1\}$, and let $\mu = \mathbf{E}\left[\sum_{i=1}^n X_i\right]$. Then $\mathbf{Pr}\left[\sum_{i=1}^n X_i < \mu/2\right] \leq e^{-\mu/12}$. Therefore, the probability that $|\text{out}_{F^*}(S)| < |\text{out}_F(S)|/32$ is at most $e^{-N \cdot 2^{i-5}/12}$. Overall, the probability that for some $S \in \mathcal{C}_i$, $|\text{out}_{F^*}(S)| < |\text{out}_F(S)|/32$ is at most:

$$(n')^{2^{i+1}} \cdot e^{-2^{i-5}N/12} < 1/(n')^2$$

since $N > 1536 \log n'$. Using the union bound over all $1 \leq i \leq \lceil \log m \rceil$, with probability at least $\frac{\lceil \log m \rceil}{(n')^2}$, for every set $S \subseteq V$, $|\text{out}_{F^*}(S)| \geq |\text{out}_F(S)|/32$. In particular, set U is $\alpha_{\text{WL}}/32$-well-linked in F^* w.h.p. This concludes the proof of Claim 5.2. As observed above, this implies that A is α-well-linked in graph H^*, thus completing the proof of Theorem 5.1.

Acknowledgement: We thank Paul Seymour for posing the question of the existence of degree-3 treewidth sparsifiers to us.

References

[1] Matthew Andrews. Approximation algorithms for the edge-disjoint paths problem via Raecke decompositions. In *Proceedings of IEEE FOCS*, pages 277–286, 2010.

[2] S. Arnborg, D. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[3] Joshua D. Batson, Daniel A. Spielman, Nikhil Srivastava, and Shang-Hua Teng. Spectral sparsification of graphs: theory and algorithms. *Commun. ACM*, 56(8):87–94, 2013.

[4] András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 47–55, New York, NY, USA, 1996. ACM.

[5] H. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[6] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423 – 434, 2009.

[7] Erin W. Chambers and David Eppstein. Flows in one-crossing-minor-free graphs. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *ISAAC (1)*, volume 6506 of *Lecture Notes in Computer Science*, pages 241–252. Springer, 2010.

[8] Moses Charikar, Tom Leighton, Shi Li, and Ankur Moitra. Vertex sparsifiers and abstract rounding algorithms. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 265–274, Washington, DC, USA, 2010. IEEE Computer Society.

[9] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *CoRR*, abs/1305.6577, 2013. Extended abstract in *Proc. of ACM STOC*, 2014.

[10] Chandra Chekuri and Julia Chuzhoy. Degree-3 treewidth sparsifiers. *CoRR*, abs/xxx, 2014.

[11] Chandra Chekuri and Alina Ene. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *Proc. of ACM-SIAM SODA*, 2013.

[12] Chandra Chekuri and Nitish Korula. A graph reduction step preserving element-connectivity and applications. In *Proc. of ICALP*, pages 254–265, 2009.

[13] Julia Chuzhoy. On vertex sparsifiers with steiner nodes. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 673–688, New York, NY, USA, 2012. ACM.

[14] Julia Chuzhoy. Routing in undirected graphs with constant congestion. In *Proc. of ACM STOC*, pages 855–874, 2012.

[15] Julia Chuzhoy and Shi Li. A polylogarithimic approximation algorithm for edge-disjoint paths with congestion 2. In *Proc. of IEEE FOCS*, 2012.

[16] Andrew Drucker. New limits to classical and quantum instance compression. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 609–618. IEEE, 2012.

[17] Matthias Englert, Anupam Gupta, Robert Krauthgamer, Harald Räcke, Inbal Talgam-Cohen, and Kunal Talwar. Vertex sparsifiers: new results from old techniques. In *Proceedings of the 13th international conference on Approximation, and 14 the International conference on Randomization, and combinatorial optimization: algorithms and techniques*, APPROX/RANDOM'10, pages 152–165, Berlin, Heidelberg, 2010. Springer-Verlag.

[18] U. Feige, M.T. Hajiaghayi, and J.R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38:629–657, 2008.

[19] Torben Hagerup, Naomi Nishimura, Jyrki Katajainen, and Prabhakar Ragde. Characterizing multiterminal flow networks and computing flows in networks of bounded treewidth. *J. Comput. Syst. Sci.*, 57, 1998.

[20] H. R. Hind and O. Oellermann. Menger-type results for three or more vertices. *Congressus Numerantium*, 113:179–204, 1996.

[21] Bart M. P. Jansen. On sparsification for computing treewidth. In *Proceedings of IPEC*, pages 216–229, 2013.

[22] David R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24:383–413, 1999.

[23] Arindam Khan, Prasad Raghavendra, Prasad Tetali, and László A. Végh. On mimicking networks representing minimum terminal cuts. *CoRR*, abs/1207.6371, 2012.

[24] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56(4):19:1–19:15, July 2009.

[25] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '12, 2012.

[26] Robert Krauthgamer and Inbal Rika. Mimicking networks and succinct representations of terminal cuts. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM*

Symposium on Discrete Algorithms, pages 1789–1799. SIAM, 2013.

[27] Alexander Leaf and Paul Seymour. Treewidth and planar minors. Manuscript, available at https://web.math.princeton.edu/ pds/papers/treewidth/paper.pdf, 2012.

[28] F. Thomson Leighton and Ankur Moitra. Extensions and limits to vertex sparsification. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 47–56, New York, NY, USA, 2010. ACM.

[29] L. Lovász. On some connectivity properties of eulerian graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 28(1-2):129–138, 1976.

[30] W. Mader. A reduction method for edge connectivity in graphs. *Ann. Discrete Math.*, 3:145–164, 1978.

[31] Konstantin Makarychev and Yury Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *FOCS*, pages 255–264. IEEE Computer Society, 2010.

[32] Ankur Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *FOCS*, pages 3–12. IEEE Computer Society, 2009.

[33] Lorenzo Orecchia, Leonard J. Schulman, Umesh V. Vazirani, and Nisheeth K. Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 461–470, New York, NY, USA, 2008. ACM.

[34] Satish Rao and Shuheng Zhou. Edge disjoint paths in moderately connected graphs. *SIAM J. Comput.*, 39(5):1856–1887, 2010.

[35] Bruce Reed. *Surveys in Combinatorics*, chapter Treewidth and Tangles: A New Connectivity Measure and Some Applications. London Mathematical Society Lecture Note Series. Cambridge University Press, 1997.

[36] N Robertson, P Seymour, and R Thomas. Quickly Excluding a Planar Graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, November 1994.

[37] Neil Robertson and P D Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, August 1986.

[38] Neil Robertson and Paul Seymour. Graph minors XXIII: Nash-williams' immersion conjecture. *Journal of Combinatorial Theory, Series B*, 100(2):181 – 205, 2010.

[39] P. Wollan. The structure of graphs not admitting a fixed immersion. *ArXiv e-prints*, February 2013.