Shallow excluded minors and improved graph decompositions

Serge Plotkin* **Satish Rao†** **Warren D. Smith†**

Abstract

In this paper we introduce the notion of the *limited-depth* minor exclusion and show that graphs that exclude small limited-depth minors have relatively small separators. In particular, we prove that for any graph that excludes K_h as a depth l minor, we can find a separator of size $O(lh^2 \log n + n/l)$. This, in turn, implies that any graph that excludes K_h as a minor has an $O(h\sqrt{n \log n})$-sized separator, improving the result of Alon, Seymour, and Thomas for the case where $h \gg \sqrt{\log n}$.

We show that the d-dimensional simplicial graphs with constant aspect ratio, defined by Miller and Thurston, exclude K_h minors of depth L for $h = \Omega(L^{d-1})$ when d is a constant. These graphs arise in finite element computations.

Our proof of separator existence is constructive and gives an algorithm to find the t-cut-covers decomposition, introduced by Kaklamanis, Krizanc, and Rao, in graphs that exclude small depth minors. This has two interesting implications. First, combination of our t-cut-cover algorithm with the results of Kaklamanis et al., gives a way to implement many finite element computations in three or more dimensions on a hypercubic parallel computer with asymptotically maximal speedup. Second, by combining the t-cut-cover algorithm with the techniques developed by Leiserson, Rao and Toledo, gives an algorithm to asymptotically improve the external memory traffic used by any out-of-core linear relaxation finite element computation in three or more dimensions.

*Department of Computer Science, Stanford University. Research supported by U.S. Army Research Office Grant DAAL-03-91-G-0102 and by a grant from Mitsubishi Electric Laboratories.

†NEC Research Institute. 4 Independence Way, Princeton, NJ 08540.

1 Introduction

1.1 Separators A separator $S \in V$ of a graph $G = (V, E)$ is a set of nodes whose removal separates the graph into connected components, each containing at most $\frac{2}{3}n$ nodes, where n is the number of the nodes in the original graph. Small cardinality separators can be used to devise efficient divide-and-conquer graph algorithms [17, 18, 2, 9].

A classical theorem of Lipton and Tarjan [17] states that any planar graph has an $O(\sqrt{n})$-size separator. Alon, Seymour, and Thomas [2] considered separators of graphs with excluded minors, where H is a minor of G if one can obtain H from G by contracting some edges and deleting others. They showed that any graph with no K_h as a minor has an $O(h^{3/2}\sqrt{n})$ separator. Since a theorem of Kuratowski [14] states that planar graphs do not have K_5 as a minor, this generalizes the Lipton-Tarjan planar separator theorem [17].

However, the result in [2] falls short of generalizing the result of Gilbert, Hutchinson, and Tarjan [10] that a graph with genus g has an $O(\sqrt{gn})$ separator; since the genus of the complete graph K_h is $(h-3)(h-4)/12$ [11] the methods of [2] only suffice to show the existence of an $O(g^{3/4}\sqrt{n})$ sized separator in genus g graphs [1].

In this paper we modify the approach of [2] to yield a number of results. One of them is that a graph that does not contain K_h as a minor has an $O(h\sqrt{n \log n})$ separator, improving the result of Alon et al. when $h = \Omega(\sqrt{\log n})$. (In the genus-$g$ case, this would show $O(\sqrt{gn \log n})$, falling short of the result of [10] by a $\sqrt{\log n}$ factor.) We note that, unlike [10, 17], our techniques do not rely on any embedding of the graph.

The main new concept in this paper is the notion of a "limited depth" minor. A minor H in a graph

[1] According to a private communication from N.Alon, the authors of [2] recently observed that by applying some of the results in the Graph Minors papers one can show the existence of a separator of size $h\sqrt{n} + c(h)$, for some (extremely fast growing) function $c(h)$ that depends only on h.

G has depth l if each node in H corresponds to a connected component in G having diameter l. [2] We describe an algorithm based on the approach of [2] that given a graph G either outputs an $O(lh^2 \log n + n/l)$-size separator or outputs a K_h-minor of G with depth l. The algorithm is fast and simple.

The importance of this new concept can be illustrated by the following simple example. Consider a graph that is a square of a bounded-degree planar graph. The theory of graph minors is of little use for such graphs, since they can have K_h as a minor for $h = \Omega(\sqrt{n})$. The Alon et al. bound [2] in this case would be $O(n^{5/4})$, which is meaningless since any graph has a linear-size separator. On the other hand, such graphs do not contain *limited depth minors* of depth $O^*(n^{1/4})$ and size $O^*(n^{1/4})$. Our results then imply that they have $O^*(n^{3/4})$ separators.[3] Actually these graphs have $O(\sqrt{n})$ sized separators; we are merely using this as an example in which our results improve on those of [2] and in which minors are less useful than limited depth minors.

An important class of graphs that exclude large shallow-depth minors are constant-dimension simplicial graphs with constant aspect ratio defined by Miller and Thurston [19]. These graphs come from d-dimensional finite element computations.

It is important to mention that our results guarantee the existence of a separator of a certain size. This should be contrasted with the results in [15, 13] where an algorithm is given that (essentially) finds a separator to within an $O(\log n)$ factor of optimal, but the optimal size is not known a priori.

1.2 Cut Covers In addition to guaranteed construction of small size separators, our techniques can be used to to produce *t-cut-covers* with guaranteed parameters. For t-cut-covers, our methods both yield an upper bound on the absolute size of covers in certain classes of graphs and the best known methods for constructing such t-cut-covers.

Roughly speaking, a *t-neighborhood cover* of a graph is a set of node subsets $\mathcal{A} = U_1, U_2, \ldots, U_k$ such that the t-neighborhood (*i.e.* all the nodes within distance t) of every node in G is contained in one of the subsets. Finding t-neighborhood covers where the subgraphs share only a small number of nodes has proved useful in a variety of contexts [4, 3, 1, 13].

Often the usefulness of t-neighborhood covers for constructing divide-and-conquer algorithms is limited by the fact that, in general, some of the subsets U_i can be very large. This lead Kaklamanis, Krizanc, and Rao [12] to define a notion of *t-cut-cover*, that combines both the concept of small cuts and the idea of t-neighborhood covers as follows. A t-cut-cover is defined as a subset of the nodes C and a t-cover $\mathcal{A} = (U_1, U_2, \ldots, U_k)$ of the graph induced by $V - C$, where no subset in $\mathcal{A}$ contains more than $2n/3$ nodes. Finding t-cut-covers for relatively large t and small $|C|$ and where the subsets share a limited number of nodes, has proved essential in several seemingly unconnected contexts, including emulating planar or nearly planar constant degree graphs in hypercubic networks [12] and reducing the memory bandwidth of iterative relaxation algorithms on graphs [16].

We show how to use our techniques for finding small separators to design improved algorithms for finding t-cut-covers for graphs that do not contain small-depth minors. This significantly extends the class of graphs for which the results of [12, 16] are applicable. For example, in contrast to the results presented in this paper, the cut-cover methods of [12] are not applicable for the graphs that arise in three or higher-dimensional finite element analysis.

1.3 Simplicial graphs Finally, we prove that constant-dimension simplicial graphs with constant aspect ratio, defined by Miller and Thurston [19], exclude L-depth K_h minors for $h = \Omega(L^{d-1})$, where d is the dimension. Such graphs appear naturally in the context of finite element analysis and describe the decomposition of the "body" into "elements". Our result that such graphs do not have large low-depth minors, together with our cut-cover techniques and the results in [16] implies better out-of-core algorithms for iterative relaxation on such graphs.

Our method is currently the most general method for finding cut-covers with specified parameters.

2 Finding small separators

In this section we define the notion of *short-depth* minors, and prove that graphs that do not have such minors have very small separators. Given a graph $G = (V, E)$ and a set $X \subseteq V$, we use $G[X]$ to denote

[2] Note that by *restricting* the class of minors that we are required to exclude we *expand* the class of graphs that satisfy the required minor-exclusion property.

[3] We say that $f(n) = O^*(g(n))$ if $f(n) = O(g(n) \log^k n)$ for some constant k.

the graph induced by the nodes in X.

Definition 2.1 Given a graph G with both node and edge weights, a *separator* is a set X where the graph $G[V-X]$ has no connected component that contains more than $2/3$-fraction of the sum of the weights of the nodes and edges in the graph. The *size* of the separator is $|X|$.

Following is a modified definition of a minor, which is crucial for stating our results. This definition is equivalent to standard definition, except that we add the notion of *depth*.

Definition 2.2 Let H and G be graphs. Suppose that for each node v of H, G contains a connected subgraph $\mathcal{A}(v)$, and for every edge uv of H there is an edge $\mathcal{E}(uw)$ in G between $\mathcal{A}(u)$ and $\mathcal{A}(v)$. If the $\mathcal{A}(v)$'s are disjoint, we call $\cup_v \mathcal{A}(v)$ an H-minor of G. In this case, we refer to the $\mathcal{A}(v)$'s as *supernodes*, and to the $\mathcal{E}(uv)$'s as *superedges*. The *depth* of the H-minor is the maximum diameter of any supernode $A(v)$.

Roughly speaking, the main theorem of this section states that a graph that does not have low-depth K_h as a minor, has provably small separators.

Theorem 2.3 Given a graph with m edges and n nodes, and integers l and h, there is an $O(mn/l)$ time algorithm that will either produce a K_h-minor of depth at most $l \log n$ or will find a separator of size at most $O(n/l + 4lh^2 \log n)$.

An appropriate choice of the depth l yields the following result, which is an improvement of the bound in [2] for $h = \Omega(\sqrt{\log n})$.

Corollary 2.4 Given a graph with m edges and n nodes, and an integer h, there is an $O(m\sqrt{n \log n})$ time algorithm that either produces a K_h-minor or finds a separator of size at most $O(h\sqrt{n \log n})$.

In order to prove Theorem 2.3, we begin with the following lemma. We use $N(S)$ to denote the set of nodes that are adjacent to the nodes in S. Roughly speaking, the lemma states that either a shallow tree containing at least one representative from each of several given subsets can be found, or that there exists a subset R of nodes that can be separated from the rest by removing a relatively small (with respect to $|R|$) number of nodes.

Lemma 2.5 Let G be a graph with n vertices, let $A_1, ...A_k$ be k subsets of the vertex set $V(G)$ and let $l \geq 1$ be an integer. Then either

1. there is a rooted tree T in G such that the depth of T is at most $4l \log n$ and such that $V(T) \cap A_i \neq \emptyset$ for $i = 1, ..., k$,
2. or there exists a $S \subset V(G)$ where
   (a) $|N(S) \cap V - S| \leq \min(|S|, |V - S|)/l$, and
   (b) $|N(V - S) \cap S| \leq \min(|S|, |V - S|)/l$.

Proof Sketch: Initialize subset R to be some node $v \in V$. Each step we will augment R by including nodes that are within distance 2 of the nodes in R. We repeat this augmentation step as long as it either causes $|R|$ to grow by a factor of $(1 + 1/l)$, or causes $|V - R|$ to shrink by a factor $(1 + 1/l)$.

Consider the set R after the execution of the last step. If R contains at least one node from each of the sets A_i, then we output a tree consisting of shortest paths from these to v. Otherwise, we output $R \cap N(R)$.

It is easy to see that after $2l \log n$ iterations, either S would contain more than n nodes, or $V - S$ would be empty, and hence the augmentation process has to stop after at most $2l \log n$ iterations. Thus, the length of a path from v to each of the representative nodes of the sets $\{A_i\}$ is bounded by $4l \log n$ as required by condition 1.

The termination condition implies that if a tree is not produced, then the size of the sets $N(R)$ and $N(N(R))$ is bounded by $\min(|R|, |V - R|)/l$, which implies conditions 2a and 2b . □

Proof of Theorem 2.3: We are given a graph G, a weight function w on its nodes and edges, and integers l, h. The algorithm maintains a three-way partition $(V_r, M, V - V_r - M)$. We will prove by induction that this partition conforms to the following invariants.

1. The set M can be partitioned into $k < h$ subsets $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$, where the size of each subset is bounded by $4lh \log n$. The diameter of each one of the subsets is bounded by $8l \log n$. For every two of these subsets there is an edge that connects a node in one to a node in the other, proving that M has K_k as a minor of depth $4l \log n$. (Subsets $\mathcal{A}_i$ correspond to supernodes of Definition 2.2.)
2. $|N(V_r) \cap (V - V_r - M)| \leq |V_r|/l$, *i.e.* V_r can be separated from the rest of the graph by removing at most $|V_r|/l$ nodes.

3. $w(G_r) \leq 2w(G)/3$, where G_r is the graph induced by V_r in G.

The algorithm proceeds in phases, where at each phase it either augments M so that it contains a larger minor, or adds nodes from M and/or nodes from $V - V_r - M$ to V_r. The algorithm terminates when the graph induced by $V - V_r - M$ contains less than 2/3-fraction of the total weight. The separator returned by the algorithm consists of $M \cup (N(V_r) - V_r)$. Observe that if the above invariants are satisfied, the size of the returned separator is bounded by $|M| + |V_r|/l \leq O(lh^2 \log n + n/l)$, as required by Theorem 2.3.

Clearly, the invariants are satisfied initially. Assume that at the beginning of some phase, M contains a K_k minor defined by supernodes $\mathcal{A}_1, ..., \mathcal{A}_k$. For all i, define A_i to be the neighbors of $\mathcal{A}_i$ that lie in $V - V_r - M$, i.e., $A_i = N(\mathcal{A}_i) - V_r - M$. If some A_i is empty then we remove the corresponding supernode from M and add it to V_r. Observe that this operation maintains the invariants.

If all of A_i's are nonempty, we apply Lemma 2.5 on the graph induced by $V - V - r - M$ in G to produce either the minimal tree T that contains a node in each A_i or a subset of nodes S.

Case 1. Tree T produced: By Lemma 2.5, the the depth of the tree is limited $4l \log n$. Notice that a minimal tree contains at most $4kl \log n$ nodes since a minimal tree consists of at most k paths of length $4l \log n$. By construction, every set $\mathcal{A}_i$ has a node adjacent to some node in the tree. Hence, if we define $\mathcal{A}_{k+1}$ as the nodes of the tree, the fact that Invariant 1 was satisfied at the start of the phase implies that it will be satisfied in the end of the phase. The only possible problem can occur if $k+1 = h$. In this case the algorithm terminates and outputs the sets $\mathcal{A}_1, \ldots, \mathcal{A}_h$ as a proof that G has a small-depth K_h minor. The set V_r is not changed and hence Invariant 3 is maintained.

Case 2. Set S returned: The algorithm adds the set S or the set $V - V_r - M - S$ to V_r, depending on which set contains the least weight. Let S' denote the set that was added to V_r. Part 2 of Lemma 2.5 implies that at most $|S'|/l$ nodes are added to $N(V_r) \cap V - V_r - M$. Since at least $|S'|$ nodes are added to V_r, Invariant 2 is maintained. Invariant 3 is maintained since, at the end of the phase, the weight of V_r bounded by $w(V_r) + (w(V) - w(V_r))/2$, which is at most $2w(V)/3$ as long as $w(V_r)$ was previously less than $w(V)/3$. (If it was not, the algorithm would have terminated in the previous iteration.) Invariant 1 is trivially maintained since the set M remains unchanged in this case. $\square$

3 Constructing cut-covers

In this section we formally define *t-cut-covers* and show how to adapt the separator algorithm described in the previous section to find good cut-covers. We use $N^t(X)$ to denote the set of nodes not in X that are within distance t from nodes in X.

In the case of cut-covers we will restrict the class of weight functions that we work with as follows.

Definition 3.1 A color or a coloring function on a graph is a $0-1$ weight function on its edges and nodes.

Definition 3.2 A *balanced* (t, s, ϵ) *cut-cover* of an N-node graph $G = (V, E)$ with a coloring function on its edges and nodes is a set $C \subseteq V$ and a set of subsets $\{V_1, ..., V_k\}$, such that

1. $|C| \leq s$,
2. $\sum_{i \leq k} w(G[V_i]) \leq (1+\epsilon) w(G)$,
3. $\max_{i \leq k} w(G[V_i]) \leq 2w(G)/3$, and
4. for all $v \in V$, there exists i such that $N^t_{G[V-C]}(v) \subseteq V_i$

where $w(G[V_i])$ is the total number of colored edges and nodes in the subgraph $G[V_i]$, and $N^t_G(v)$ denotes the set of nodes that are within distance t of v in G.

As we stated in the introduction, for applications we will need to find cut-covers that have relatively large values of t, and relatively small s and ϵ. For cut-covers, we will now state a more general theorem that shows how to find the multi-colored cut-covers required for the applications of [12, 16]. We do not show how to find multicolored separators since Theorem 2.3 can be used a subroutine to find such separators, as described in [6, 7]. In the case of cut-covers, it appears that an algorithm that finds a single-colored cut-cover cannot easily be used to find a multicolored cut-cover, and hence we will directly prove the following multicolored version.

Theorem 3.3 Given a graph with m edges and n nodes with q colors on its nodes and edges, and integers h, t, and $l > 2qt$, there is an $O(mn/l)$ time algorithm that will produce a set C and a set of subgraphs that is a balanced $(t, O(q^2 lh^2 \log n), 2qt/l)$ cut-cover with respect to each of the q colors ignoring the other $q-1$ colors.

The main idea in proving the above theorem is to modify the algorithm used in the proof of Theorem 2.3. In that proof, the separator was constructed from nodes in some K_k-minor and neighbors of certain sets of nodes in the remaining graph. Here, we find a cut-cover $C, \{S_1, ..., S_k\}$ where the nodes in the minor correspond to the set C, and t-neighborhoods of the above mentioned sets correspond to the S_i's. Complications arise, however, since certain nodes appear in many S_i's. We must ensure that the total weight $\sum_i w_j(G[S_i])$ for any color class j is not substantially increased during this process. We proceed with proving the following generalization of Lemma 2.5.

Lemma 3.4 Let $G = (V, E)$ be a graph with n vertices, let $A_1, ...A_k$ be k subsets of V, let $w_1, ..., w_q$ be coloring functions on the edges and nodes of G, and let t and $l \geq 4t$ be integers. Then either

1. there is a tree T in G such that $|V(T)| \leq 8(q+1)kl\log n$ such that $V(T) \cap A_i \neq \emptyset$ for $i = 1, ..., k$,
2. or there exists an $S \subset V(G)$ where
   (a) $|N^t(S) \cap V - S|$ is at most $t\min(|S|, |V-S|)/l$, and
   (b) $|N^t(V-S) \cap S|$ is at most $t\min(|S|, |V-S|)/l$.
   (c) for each $i \in \{1, ..., q\}$, $w_i(N^t(S) \cap V - S)$ is at most $tw_i(S)/l$,
   (d) for each $i = \{1, ..., q\}$, $w_i(N^t(V-S) \cap V)$ is at most $tw_i(V-S)/l$,

where $w_i(S)$ is the total number of edges and nodes that are colored with i in the subgraph induced by S in G.

Proof Sketch: We initialize a subset R to be some node v in G. We augment the subset R by including nodes that are within distance $2t$ of S. If the size of R or if $w_i(R)$ for some i grows by a factor of $(1+t/l)$ or the size of $V-R$ or if $w_i(V-R)$ for any i shrinks by a factor of $(1+t/l)$, we continue this process. Otherwise, the process terminates.

As in the proof of lemma 2.5, we check if the final subset R contains at least one node from each of the A_i's. If it does, then we output the tree consisting of shortest paths from each of the nodes in the A_i's to v. If R does not contain a node from each of the A_i's, then we output $N(R) \cup R$.

Observe that the process terminates in fewer than $4l(q+1)\log n/t$ iterations. This follows since at least one color or the set of nodes in R or not in R change by a factor of $(1+t/l)$ in every iteration. Thus, if there are more than $4l(q+1)\log n$ iterations then the size of one of the color classes or the size of the set of nodes can be used to produce a contradiction. That is, either the number of nodes will grow to be larger than n or the number of colored nodes or edges in some color class will grow to be larger than n^2.

Thus, the paths to v that are used by the k representatives of the A_i's have length at most $8(q+1)l\log n$. We conclude that the total size of the tree is $8k(q+1)l\log n$ as required by condition 1. Also, by considering the termination condition of the process used to construct R, it is easy to show that conditions 2a - 2d are satisfied when a subset is output. □

Now we can proceed by using an algorithm similar to that of the previous section to prove the following theorem.

Theorem 3.5 Given a G, with q colors on its nodes, and integers h, t, and $l > 2qt$, we can find a balanced $(t, O(qlh^2\log n), t/l)$-cut-cover for the first color that does not increase the total number of edges and nodes that are colored with any other color by more than a factor of $(1+t/l)$.

Proof Sketch. We proceed as in the algorithm of the previous section except that here we produce a set of subsets $\{S_1, ..., S_k\}$ during the course of the algorithm as follows. Each time a set S is produced using the lemma 3.4, we produce a set S' as in Case 2 of the proof of Theorem 2.3. We add the $N^t(S')$ to the set of subsets and proceed as before.

Now, using Lemma 3.4, it is possible to prove that, upon termination, the cut-set M and subsets $\{S_1, ...S_k\}$ form a balanced $(t, O(qh^2l\log n), t/l)$-cover of G for one of the colors. Moreover, the weight $\sum_i w_j(G[S_i])$ is not much larger than $w_j(G)$ for any color j. □

We can use the above theorem to produce a balanced $(t, O(q^2lh^2\log n), 2qt/l)$-cut-cover for all the colors simultaneously as required in Theorem 3.3. Roughly speaking, the idea is to apply the theorem above to find a cut cover for the first color, then apply the above theorem to the $\cup G[S_i]$ for the second color, add the resulting "cut" nodes to the set of cut nodes from the previous iteration, and so on. Each iteration may increase the total number of nodes (or colored nodes and edges for any of the q colors) by at most $(1+t/l)$, and may add lh^2 nodes to C. This leads to a balanced

$(t, O(q^2 lh^2 \log n), 2qt/l)$ cut-cover, (since $(1+t/l)^q \leq 1+2qt/l$ when $l > 2qt$) as required in Theorem 3.3.

4 Applications

We discuss two applications of our cut-covers result in this section. We refer the reader to [7, 18, 2] for numerous separator-related applications. The first application is developed in [12], where they show that n-node graphs that have appropriate cut-covers can be emulated in n-node hypercubes with constant slowdown or in n-node butterflies with $O(\log^* n)$ slowdown. They show that this can be done even though there is no low dilation embedding of these graphs into hypercubes or butterflies.[4] Roughly speaking, the idea is that a cut-cover $(C, \{S_1, ..., S_k\})$ decomposes graphs into subgraphs (induced by the S_i's) that can run for t steps with minimal direct communication (only among the neighbors of C) between the subgraphs. The following corollaries follow from the work of [12] and our theorem 3.3.

Corollary 4.1 An N-node butterfly can emulate any $T > \log N$ steps of an N-node graph G in $O(dTO(\log^* N))$ steps if G has maximum degree d and no k-node subgraph contains a $K_{\log^{O(1)} k}$-minor of depth less than $\log k$.

Corollary 4.2 An N-node hypercube can emulate $T > \log N$ steps of an N-node graph G in time $O(dT)$ if G has maximum degree d that does not contain $K_{\log^{O(1)} N}$ of depth less than $\log n$.

The second application is that of reducing the paging required to perform a linear relaxation algorithm on a graph. An example of this is the repeated multiplication of a vector x by a sparse matrix A where the pattern of nonzeros in A defines the underlying graph. When the entire vector x (or the entire matrix A) is too large to be in primary memory, multiplying a vector x repeatedly by the matrix A causes the entire vector (and the entire matrix A) to be paged in for each matrix-vector multiply.[5] It is shown in [16] that if the underlying graph has appropriate cut-covers then the amount of paging can be greatly reduced. Their idea is to introduce a variable for each node of C for each timestep of the relaxation algorithm and maintain linear dependencies among the vertices of C, thereby allowing each subgraph of the t-cover to be iteratively relaxed for t timesteps without external communication. The following corollary follows from their work and Theorem 3.3.

Corollary 4.3 A computer with M words of primary memory can perform w steps of a linear relaxation scheme on a graph $G = (V, E)$ that does not contain a K_h minor of depth less than l where $l \geq 4t \log n$, with $O(w|E| + |V||E|/l^2)$ computation and $O(|E|(w/\min(\sqrt{w}, \sqrt{(M/lh^2)})) + |E||V|/l^2)$ secondary memory references.

We remark that the $|E||V|/l^2$ term in the corollary above arises from the time required for finding cover-cuts. This work needs only be performed once for any graph regardless of the actual values in the matrix A that corresponds to the graph. Thus, if many relaxation schemes are computed on a single graph, this cost is only incurred once. In addition, [16] give a method for reducing the memory bandwidth for the conjugate gradient method where the operations are not all linear and the interactions between adjacent timesteps are not defined exclusively by the graph underlying a matrix. Again, they use cut-cover decompositions of the graph.

5 Simplicial graphs exclude shallow minors

In this section, we show that the graphs that arise naturally from finite element computations in 3 or higher dimensions exclude shallow minors. In particular, we will show that the d-dimensional simplicial graphs of bounded "aspect ratio" as defined by Miller and Thurston [19] exclude K_h as a depth L minor if $h = \Omega(L^{d-1})$ for constant d. While Miller and Thurston already had shown that such graphs have small separators, our statement leads to good cut-covers for such graphs. For example, we can use our techniques to efficiently perform out-of-core finite element computations on such graphs as in [16] or to efficiently simulate simplicial graphs in hypercubic parallel computers [12].

A *k-dimensional simplex*, or *k-simplex*, is the convex hull of $k+1$ affinely independent points in $\Re^d$ space. A d-dimensional simplicial complex is a collection of k-simplices, $k \leq d$, closed under subsimplex and intersection.

For example, a 3-dimensional simplicial complex is a collection of cells (3-simplexes), faces (2-simplices), edges (1-simplices), and vertices(0-simplices). A *d-dimensional simplicial graph* is the collection of edges

[4] In fact, there is a lower bound of $\Omega(\log n)$ on the dilation of graphs in this class.

[5] We remark that precomputing A^k and then multiplying by x may not be wise since A^k is likely to be dense.

(1-simplices) and vertices (0-simplices) in a simplicial complex in d-dimensions.

The *aspect ratio* of a body, as defined by Miller and Thurston, is its diameter divided by the dth root of its volume. The volumes of a regular d-simplex, d-cube, and d-ball of unit diameter are respectively $2^{-d/2}\sqrt{d+1}/d!$, $d^{-d/2}$, and $2^{-d}\pi^{d/2}/(d/2)!$. Thus the aspect ratios of a d-simplex, d-cube and d-ball are $\alpha_s = 2^{1/2}(d!)^{1/d}(d+1)^{-1/(2d)}$, $\alpha_c = \sqrt{d}$, and $\alpha_b = 2\pi^{-1/2}(d/2)!^{1/d}$, which behave like $\sqrt{2}d/e$, $\sqrt{d}$, and $\sqrt{2d/(e\pi)}$ respectively for d large.

A simplicial graph of aspect ratio α means a simplicial graph that comes from a complex in which every d-simplex has aspect ratio at most α.

Following is the main theorem of this section.

Theorem 5.1 A simplicial graph in d-dimensions with aspect ratio α does not contain K_h as a minor of depth L if $h > L^{d-1}(\frac{128e\pi\alpha^2}{d})^{d/2+O(1)}$.

Before proceeding we remark that this theorem is not known to be tight. That is, there is no known d-dimensional simplicial graph of bounded aspect ratio that contains a minor of depth L of size $\Omega(L^{d-1})$ when d is fixed. Our best construction has $\Omega(L^{d/2})$. Closing this gap seems to be an interesting open problem. Another interesting question is whether the d-dimensional simplicial graphs of bounded aspect ratio can be *characterized* by exclusion of *low-depth* minors. (Certainly, no ordinary minor can be excluded from such graphs.)

Proof of Theorem 5.1: Roughly speaking, the outline of the proof is as follows. First, we define the notion of a "round body" and bound the number of round bodies that can touch each other in d dimensions. Then we show that given a simplicial graph with a large shallow minor, we can find many round bodies that touch each other in d-space. From this we deduce that simplicial graph cannot have too large a minor that is shallow.

A *body* in d-dimensions is a connected compact set with nonzero measure. The *diameter* of a d-dimensional body is the maximum distance between any pair of points in it. The *radius* of a d-dimensional body is the radius of the smallest d-sphere which encloses it. A body is γ-*round*, $0 \le \gamma \le 1$, if it contains at least a fraction $\ge 2^{-d}\gamma$ of the volume in any sphere whose center is a point in the body and that does not contain the entire body.

For example, a d-ball is 1-round; a regular d-simplex is only $\sqrt{d+1}(2/\pi)^{d/2}(d/2)!/d!$-round (this quantity, call it γ_s, behaves something like $(\pi d/e)^{-d/2}$ for large d). A d-simplex with aspect ratio α is $(2/\pi)^{d/2}(d/2)!(d!)^{1/(d-1)}(\frac{\sqrt{2}}{\alpha})^{d^2/(d-1)}(d+1)^{-1/(2d-2)}$-round. This quantity behaves like $(\frac{2d}{e\pi\alpha^2})^{d/2+O(1)}$ for large d.

Two bodies *touch* if they share a point. (Note that this definition implies that a body "touches" itself.) The *clique number* of a set of bodies whose interiors are disjoint is the maximum number of bodies that all touch each other.

Theorem 5.2 If there is a set of N interior-disjoint γ-round bodies in d dimensions such that each body touches the body with the smallest radius, then $N \le 4^d/\gamma$.

Proof: Let the body B_0 have the smallest radius and let its minimal enclosing ball have unit radius, WLOG. Define a concentric sphere C whose radius is 2.

We define a sphere S_i for each body, B_i, that touches B_0, whose center is a point of contact between B_i and B_0 and whose radius is 1. Since each B_i has radius of at least 1, S_i does not entirely contain B_i. Furthermore, each S_i is entirely contained in C. Thus, by the definition of γ-round, each body B_i that touches B_0 uses up at least a fraction $4^{-d}\gamma$ of the volume of C. □

Lemma 5.3 Given a simplicial graph in d-dimensions with aspect ratio α, and a K_h-minor of depth L, there is a set $\mathcal{B}$ of h interior-disjoint bodies in d dimensions, each $L^{1-d}(\frac{d}{8e\pi\alpha^2})^{d/2+O(1)}$-round, so the clique number of $\mathcal{B}$ is h.

Proof Sketch: We start with the embedding of the simplicial graph, G, into d dimensions. We then partition the complex into disjoint "territories" where territory T_i contains, and is "owned" by edge (or 1-simplex) i. We do this in such a way that each edge in a simplex owns a volume-fraction of that simplex proportional to its length to the power d. Since the simplicial graph had aspect ratio of α each edge whose embedded length is ℓ has a territory whose volume is at least $\frac{2}{(d+1)d}\gamma_s(\frac{\alpha_s}{\alpha})^{d^2/(d-1)}$ (which is $(\frac{2d}{e\pi\alpha^2})^{d/2+O(1)}$) as large as the volume of a d-ball of diameter ℓ.

We form bodies that correspond to nodes in the minor as follows. We number the minor nodes 1 to h. Since G contains K_h as a depth L minor, node i corresponds to a set of nodes in G that has a spanning tree of depth L. The body for node i is the union of all the territories corresponding to edges in its spanning tree, plus half of the territories corresponding to edges between the spanning tree for minor node i and any spanning tree for a minor node j where $j \neq i$.

By the definition of minor and construction, we know that the clique number of these bodies is h. Furthermore, the bodies are internally disjoint since the territories are.

To finish, we prove that each body is γ-round for large enough γ follows. Given a point in the body and a ball of radius R that does not contain the entire body, there is a path from the point through the body to the boundary ("the sphere") of the ball using at most $2L+1$, territories. And only the last territory in the chain can intersect the sphere.

In the event that the last territory is big, roundness is obvious since each territory is round. In the event it is small compared with the radius of the sphere, then the volume contained in the territories intersected with the ball is at least a fraction $(2L)^{1-d}2^{-d}\frac{2}{(d+1)d}\gamma_s(\frac{\alpha_s}{\alpha})^{d^2/(d-1)}$ of the volume of the sphere, since $\sum_{i=1}^{N} x_i^d \geq ((\sum_{i=1}^{N} x_i)/N)^d N$, and in this case $N \leq 2L$ and $\sum_{i=1}^{N} x_i \geq R/2$. Thus each body is $L^{1-d}(\frac{d}{8e\pi\alpha^2})^{d/2+O(1)}$-round. $\square$

Notice that if the clique number of a collection of bodies is h that the number of bodies that touch the smallest body is at least h. Theorem 5.1 then follows from theorem 5.2.

Acknowledgements We would like to thank Charles Leiserson and Sivan Toledo for suggesting the concept of low-depth minors.

References

[1] A. Agrawal, P.. Klein, S. Rao, and R. Ravi. Approximation through multicommodity flow. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 726–737, October 1989.

[2] N. Alon, P. Seymour, and Thomas R. A separator theorem for graphs with an excluded minor and its applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 293–299, May 1990.

[3] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Fast constructions of sparse neighborhood covers. Unpublished Manuscript.

[4] B. Awerbuch and D. Peleg. Sparse partitions. In *31st Symposium on Foundations of Computer Science*, pages 503–513, 1990.

[5] S. Bhatt, F. Chung, J. Hong, T. Leighton, and A. Rosenberg. Optimal simulations by butterfly networks. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 192–204, May 1988.

[6] S.N. Bhatt and F.T. Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2), 1984.

[7] F. R. K. Chung. Separator theorems and their applications. In B. Korte, L. Lovasz, H. J. Promel, and A. Schriver, editors, *Paths, Flows, and VLSI-Layout*, pages 17–34. Springer–Verlag, New York, 1990.

[8] F.R.K. Chung and R.L. Graham. On universal graphs. *Annals of New York Academy of Science*, 319:136–140, 1979.

[9] G. N. Fredrickson and R. Janardan. Separator-based strategies for efficient message routing. In *27th Annual Symposium on Foundations of Computer Science, IEEE*, pages 428–437, 1986.

[10] John R. Gilbert, Joan P. Hutchinson, and Robert E. Tarjan. A separator theorem for graphs of bounded genus. *J. Algorithms*, 5:375–390, 1984.

[11] G.Ringel and J.W.T.Youngs. Solution of the heawood map coloring problem. *Proc. Natl. Acad. Sci. USA*, 60:438–445, 1968.

[12] C. Kaklamanis, D. Krizanc, and S. Rao. New graph decompositions and fast emulations in hypercubes and butterflies. In *Proceedings of the 5th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 325–334, June 1993.

[13] P.. Klein, S. Rao, and S. Plotkin. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 682–690, May 1993.

[14] Kazimir Kuratowski. Sur le probléme des courbes gauches en topologie. *Funda. Math.*, 15:271–283, 1930.

[15] F. T. Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *28th Annual Symposium on Foundations of Computer Science, IEEE*, pages

256–269, 1988.

[16] C. E. Leiserson, S. Rao, and S. Toledo. Efficient out-of-core algorithms for linear relaxation.

[17] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, April 1979.

[18] Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, August 1980.

[19] G. Miller and W. Thurston. Separators in two and three dimensions. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 300–309, May 1990.