Decomposition, Approximation, and Coloring of Odd-Minor-Free Graphs

Erik D. Demaine* MohammadTaghi Hajiaghayi† Ken-ichi Kawarabayashi‡

Abstract

We prove two structural decomposition theorems about graphs excluding a fixed odd minor H, and show how these theorems can be used to obtain approximation algorithms for several algorithmic problems in such graphs. Our decomposition results provide new structural insights into odd-H-minor-free graphs, on the one hand generalizing the central structural result from Graph Minor Theory, and on the other hand providing an algorithmic decomposition into two bounded-treewidth graphs, generalizing a similar result for minors. As one example of how these structural results conquer difficult problems, we obtain a polynomial-time 2-approximation for vertex coloring in odd-H-minor-free graphs, improving on the previous $O(|V(H)|)$-approximation for such graphs and generalizing the previous 2-approximation for H-minor-free graphs. The class of odd-H-minor-free graphs is a vast generalization of the well-studied H-minor-free graph families and includes, for example, all bipartite graphs plus a bounded number of apices. Odd-H-minor-free graphs are particularly interesting from a structural graph theory perspective because they break away from the sparsity of H-minor-free graphs, permitting a quadratic number of edges.

1 Introduction

Decomposition or partitioning of graphs into smaller pieces is a fundamental way to design graph algorithms. One of the most famous such decompositions is the divide-and-conquer separator decomposition for planar graphs of Lipton and Tarjan [LT80], which has been generalized to arbitrary graphs via sparsest cut [LR99, ARV04]. These decompositions are based on finding relatively small cuts in the graph to minimize the interaction between the pieces. To make the pieces relatively small, the decompositions cut the graph into many pieces.

A different kind of decomposition that has received more attention recently is to partition the graph into a small number of computationally simpler (but not necessarily small) pieces, without much regard to the interaction between the pieces. For example, many optimization problems can be solved exactly on graphs of bounded treewidth; what graphs can be partitioned into a small number k of bounded-treewidth pieces? In many cases, each piece gives a lower/upper bound on the optimal solution for the entire graph, so solving the problem exactly in each piece gives a k-approximation to the problem. Such a decomposition into bounded-treewidth graphs would also be practical, as many NP-hard optimization problems are now solved in practice using dynamic programming on low-treewidth graphs; see, e.g., [Bod05, Ami01, Tho98]. Recently, this decomposition approach has been successfully used for graph coloring, which is inapproximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ unless P = NP [Zuc07], yet has a 2-approximation in any minor-closed graph family by this approach [DHK05]. Refinements of this approach have led to PTASs for many graph problems [DHK05, Bak94, Epp00].

Decomposition of graphs into computationally simpler pieces goes back to the work of Nash-Williams in 1964 [NW64] and of Chartrand et al. [CGH71]. In particular, Chartrand et al. [CGH71] conjectured in 1971 that any planar graph can have its edges partitioned into two pieces, each inducing an outerplanar graph. This conjecture was only just proved by Goncalves [Gon05], together with a linear-time algorithm for computing the decomposition. In particular, this result establishes that any planar graph can be decomposed into two graphs of treewidth 2, a result proved earlier [Ked96]. Ding et al. [DOSV00] proved that every bounded-genus graph can be decomposed into two graphs of bounded treewidth (where the bound depends on the genus). Thomas [Tho95] conjectured that every graph excluding a fixed minor has such a decomposition. This conjecture has been proved in two papers [DDO$^+$04, DHK05]; the latter proof is both simpler and algorithmic. A recent extension of this result [DHM07] allows the bounded-treewidth pieces to be formed by contractions instead of deletions/subgraphs.

In this paper, we generalize the decomposition result to *odd-minor-free graphs*: every graph excluding a fixed odd minor can be decomposed into two bounded-treewidth graphs. The family of odd-H-minor-free graphs is strictly more general than H-minor-free graphs for any graph H; for example, it includes all complete bipartite graphs $K_{n/2,n/2}$. In interesting contrast, the Lipton-Tarjan separator decom-

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA. edemaine@mit.edu

†AT&T Labs — Research, 180 Park Ave., Florham Park, NJ 07932, USA, hajiagha@research.att.com. This work was performed while the author was at Carnegie Mellon University.

‡National Institute for Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. k_keniti@dais.is.tohoku.ac.jp

position (and thus its algorithmic consequences) cannot be generalized to odd-minor-free graphs, because small separators do not exist. Another contrast between odd-H-minor-free graphs and H-minor-free graphs is that the former can have a quadratic number of edges, while the latter are always sparse. Odd-minor-free graphs have been considered extensively in the graph theory literature (see, e.g., [Gue01, GGG$^+$04, Gue05, JT95]) and recently in theoretical computer science [KM06].

We prove our result by generalizing a structural result that is the heart of Graph Minor Theory [RS03], which has many algorithmic applications [Gro03, DFHT05, DH05, DHK05], to the case of odd-minor-free graphs. Specifically, we prove that every odd-minor-free graph can be decomposed into a clique-sum of "almost-bipartite" graphs and graphs that are "almost-embeddable" into bounded-genus surfaces. (In the original Graph Minor version, only graphs of the second type exist.) Our primary challenge in this result is that odd-minor-free graphs can be dense, and this density may be equally spread throughout the graph, so we cannot hope to find small separations to split the graph into pieces as in previous decomposition theorems. Nonetheless, we show how to decompose odd-minor-free graphs, by showing a connection to the existence of clique minors; this algorithmic technique may be useful for future structural algorithms on dense graphs.

All of our results are algorithmic: the decompositions can be computed in polynomial time. Our decompositions can also be used to obtain a variety of approximation algorithms for problems on odd-minor-free graphs. In particular, our decomposition into two bounded-treewidth graphs (Theorem 1.1 below) immediately leads to a 2-approximation for many NP-complete problems in odd-minor-free graphs. For example, our 2-approximation for graph coloring in odd-K_k-minor-free graphs improves the recent $O(k)$-approximation algorithm [KM06] and generalizes the 2-approximation for K_k-minor-free graphs [DHK05]. We also believe that combining our clique-sum decomposition (Theorem 1.2 below) with Grohe's technique [Gro03] leads to PTASs for maximum independent set, maximum clique, and minimum vertex cover. Grohe's technique for minor-free graphs uses the property that removing small sets of vertices in such graphs results in components with treewidth bounded in terms of diameter; this property simply for odd-minor-free graphs (e.g., the complete bipartite graph has diameter 2 but huge treewidth even after removing small sets of vertices), making this generalization particularly interesting.

The classes of graphs excluding a fixed minor H have already proved extremely useful and powerful, with applications in many areas such as network design and compact routing; see, e.g., [AG06, Tho04]. Our thesis is that most algorithmic results that have been obtained for both H-minor-free graphs and bipartite graphs can be generalized to odd-H-minor-free graphs. Our results provide the first step in this direction by providing new algorithmic structural tools for odd-minor-free graphs. In particular, the minor-free analogs of our decomposition results have found many applications in just the past few years—see, e.g., [BBCH07, APS07, ASS07, Kle05, DFHT05, DHM]—and we analogously expect our new tools to enable several new algorithmic results for odd-minor-free graphs.

Our immediate goal is not to develop practical algorithms: it is not clear where odd-minor-free graphs would arise in practice, and even most algorithms for H-minor-free graphs are currently impractical because of huge bounds from Graph Minors. Rather, the goal of this paper is to improve our algorithmic structural understanding, in particular powerful decompositions enabling efficient approximation algorithms, beyond standard graph families.

1.1 Odd-Minor-Free Graphs and Their Importance.

Recall that a graph H is a *minor* of G if H can be obtained by contracting and deleting edges in G. Equivalently, H is a minor of G precisely if there are $|V(H)|$ vertex-disjoint trees in G, one tree T_v for each vertex v of H, such that for every edge $e = \{v, w\}$ in H there is an edge $\hat{e}$ in G connecting the two corresponding trees T_v and T_w. Now H is an *odd minor* of G if, in addition, all the vertices of the trees can be two-colored in such a way that (1) the edges within each tree T_v in G are bichromatic, while (2) the edge $\hat{e}$ connecting trees T_v and T_w in G corresponding to each edge $e = \{v, w\}$ of H is monochromatic. In particular, the class of odd-H-minor-free graphs (excluding a fixed graph H as an odd minor) is more general than the class of H-minor-free graphs (excluding a fixed graph H as a minor).

Indeed, the class of odd-H-minor-free graphs is strictly more general: the complete bipartite graph $K_{n/2,n/2}$ certainly contains a K_k minor for $k \leq n/2$, but on the other hand, it does not contain K_k as an odd minor for $k \geq 3$. In fact, any K_k-minor-free graph G is $O(k\sqrt{\log k})$-degenerate, i.e, every induced subgraph has a vertex of degree at most $O(k\sqrt{\log k})$; see [Kos84, Tho01]. Thus, any K_k-minor-free graph G has $O(k\sqrt{\log k}n)$ edges. On the other hand, some odd-K_k-minor-free graphs such as $K_{n/2,n/2}$ may have $\Theta(n^2)$ edges.

This contrast seems to make a big difference between minor exclusion and odd minor exclusion. On the other hand, as we shall see later, odd minors are actually motivated by Graph Minor Theory and structural graph theory, and many researchers believe that there is an analogue of Graph Minor Theory for the case of odd minors. In addition, odd minors have an intriguing connection to the well-known conjecture of Hadwiger [Had43], detailed below.

Connections to Maximum Cut. Odd minors play an important role in the field of discrete optimization. A long-standing area of interest in this field is finding conditions under which a given polyhedron has integer vertices, so that integer optimization problems can be solved as linear programs. In the case of a particular set-covering formulation of the maximum-cut problem, there is a structural characterization based on excluding odd minors.

Specifically, consider the following problem. A *signed graph* is a pair (G, Σ), where $G = (V, E)$ is an undirected graph and $\Sigma \subseteq E$. A circuit in (G, Σ) is *odd* if it contains an odd number of edges in Σ. An *odd circuit cover* is a set of edges intersecting all odd circuits. The problem of finding an odd circuit is more general than finding a maximum cut in G (which is NP-hard); see [GP82, Sch02]. The linear relaxation of a natural integer programming formulation of the problem is the following:

1. $x(e) \geq 0$, for each edge e in G;
2. $\sum_{e \in C} x(e) \geq 1$, for each odd circuit C in G.

A signed graph (G, Σ) is *weakly bipartite* if each vertex of the polyhedron determined by these constraints is integral. Weakly bipartite graphs are important because we can find a maximum-capacity cut in such graphs in polynomial time by solving the above linear program via the ellipsoid method.

Guenin [Gue01] characterized weakly bipartite graphs in terms of forbidden odd minors, for which he won the Fulkerson prize in 2003. The characterization says that a signed graph (G, Σ) is weakly bipartite if and only if G does not contain an odd K_5 minor. This theorem proves a special case of a well-known conjecture of Seymour [Sey77]. It also generalizes a result of Seymour [Sey77] that a signed graph (G, Σ) is "strongly bipartite" if and only if G does not contain an odd K_4 minor. (For the definition of strongly bipartite graphs, we refer the reader to [Gue01, Sch02].) Guenin's result [Gue01] has motivated several remarkable subsequent papers; see [GG02, Sch02].

Thus, as we see, odd minors are useful for proving structure theorems in discrete optimization.

Connections to Hadwiger's Conjecture. Odd-minor-free graphs has also been used to generalize Hadwiger's conjecture, considered by many as the deepest open problem in graph theory. Hadwiger's conjecture is a far-reaching generalization of the Four Color Theorem [AHK77, AH77, RSST97]. It states that every H-minor-free graph has a vertex coloring with $|V(H)| - 1$ colors. Hadwiger [Had43] posed this problem in 1943, and proved the conjecture for $|V(H)| \leq 4$. The case $|V(H)| = 5$ is equivalent to the Four Color Theorem, as proved by Wagner in 1937 [Wag37], and therefore is also true [AHK77, AH77, RSST97]. The case $|V(H)| = 6$ was proved by Robertson, Seymour, and Thomas [RST93], also using the Four Color Theorem. All cases $|V(H)| \geq 7$ remain unsolved. The only known results for $|V(H)| = 7$ are that any 7-colorable graph has K_7 or $K_{4,4}$ as a minor [KT05] and has K_7 or $K_{3,5}$ as a minor [Kawb], whereas Hadwiger's conjecture suggests that the graph should always have a K_7 minor. The best general upper bound is that every H-minor-free graph has a vertex coloring with $O(|V(H)|\sqrt{\lg |V(H)|})$ colors, which follows immediately from bounds on the average degree of a vertex in an H-minor-free graph; see, e.g., [Kos84, Tho01]. Thus, Hadwinger's conjecture is not resolved even up to constant factors. On the other hand, a 2-approximation for graph coloring in H-minor-free graphs was recently obtained in [DHK05].

In 1993, Gerards and Seymour [JT95, p. 115] conjectured a substantially stronger form of Hadwiger's conjecture in terms of odd minors: every odd-K_k-minor-free graph has a vertex coloring with $k - 1$ colors. The cases $k \leq 3$ are straightforward; for example, the case $k = 3$ states that a graph with no odd cycles is 2-colorable. The case $k = 4$ was proved by Catlin in 1978 [Cat78]. Recently, Guenin [Gue05] announced a solution for the $k = 5$ case, which certainly implies the Four Color Theorem because a graph with no K_5 minors also has no odd K_5 minors. All cases $k \geq 6$ remain unsolved.

As with Hadwinger's conjecture, it is not even known whether there is a constant $c \geq 1$ such that any ck-colorable graph contains an odd K_k minor. The best general upper bound is that every odd-K_k-minor-free graph has a vertex coloring with $O(k\sqrt{\lg k})$ colors, which follows from bounds on the average vertex degree in such graphs by Geelen et al. [GGG+04]. On the other hand, Kawarabayashi [Kawa] gave an algorithm for any fixed k that either (1) colors a given graph with ck colors, (2) finds an odd K_k minor in the graph, or (3) constructs a counterexample to the weaker conjecture that every ck-colorable graph contains an odd K_k minor. Here, $c = 2496$. The proof also implies that this weaker conjecture has only finitely many minimal counterexamples, and that any such counterexample is $k/2497$-connected (improving on 4-connectivity as proved by Guenin [Gue05]).

One of the major consequences of this work is an $O(k)$-approximation for graph coloring in odd-K_k-minor-free graphs [KM06]. In this paper, we improve this approximation factor to 2, independent of k. This result generalizes the previous 2-approximation algorithm for K_k-minor-free graphs [DHK05] to a much more general graph family.

1.2 Our Results.

One main result of this paper is the following decomposition into two bounded-treewidth graphs, which we also show is tight in a certain sense (see Section 3):

THEOREM 1.1. *For every positive integer k, there is a constant c_k such that, for every odd-K_k-minor-free graph G, the vertices of G can be partitioned into two sets such that each set induces a graph of treewidth at most c_k. Furthermore, such a partition can be found in polynomial time.*

Because graph coloring can be solved in polynomial time on graphs of bounded treewidth, we obtain a 2-approximation algorithm for graph coloring:

COROLLARY 1.1. *There is a polynomial-time 2-approximation algorithm for graph coloring in graphs that exclude a fixed odd minor.*

This result significantly generalizes the previous 2-approximation algorithm for graph coloring in graphs excluding a fixed minor [DHK05]. Furthermore, our result improves on the recent $O(k)$-approximation algorithm for graph coloring in odd-K_k-minor-free graphs [KM06]. This approximation factor is particularly impressive given that graph coloring is one of the hardest problems to approximate. In general graphs, it is inapproximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$, unless P = NP [Zuc07]. Even for 3-colorable graphs, the best approximation algorithm achieves a factor of $O(n^{0.2072})$ [Chl07]. In planar graphs, the problem is $4/3$-approximable, and that is the best possible unless P = NP, essentially because all planar graphs are 4-colorable. In contrast, H-minor-free graphs (or even bounded-genus graphs) are not $O(1)$-colorable for a constant independent of H (or genus).

The approximation approach of Corollary 1.1 applies more generally to many other optimization problems, including a variety of hereditary maximization problems (as defined in [DHK05, Section 3.3] and [Yan78]) and some minimization problems such as minimum color sum [BNBH+98, FK98, HK02]:

COROLLARY 1.2. *There is a polynomial-time 2-approximation algorithm for minimum color sum and for finding a maximum (weighted) induced subgraph with a given hereditary property in graphs that exclude a fixed odd minor. In particular, the hereditary maximization problems include independent set and finding the maximum induced subgraph that is chordal, acyclic, without cycles of a specified length, of maximum degree $r \geq 1$, bipartite, a clique, or planar.*

Our decomposition of Theorem 1.1 is based on the following structural result about odd-H-minor-free graphs, which is our second main result. This decomposition generalizes a similar structure theorem for H-minor-free graphs by Robertson and Seymour [RS03], which is the heart of Graph Minor Theory, and which has been used recently to obtain many algorithmic consequences in polynomial-time approximation schemes, subexponential fixed-parameter algorithms, approximating treewidth, approximating grid minors, and half-integrality of multicommodity flow [Gro03, DFHT05, DH05, DHK05]. For the definition of *h-almost-embeddable graphs*, we refer the reader to Section 2. Let us remark that h-almost-embeddable graphs involve bounded genus graphs.

THEOREM 1.2. *For any fixed k, there is a constant h such that any odd-K_k-minor-free graph G can be obtained by clique-sums (at most h-sums) of the following two types of graphs:*

1. *bipartite graphs together with at most h apex vertices; and*

2. *h-almost-embeddable graphs.*

Moreover, if $G_1 \oplus G_2$ is a clique sum and G_2 is a child of G_1, then $G_1 \cap G_2$ is contained in the apex vertex set of G_2. Furthermore, if G_1 is a bipartite graph W together with at most h apex vertices, then $|G_2 \cap W| \leq 1$. Finally, the decomposition can be computed in polynomial time.

This result is proved in Section 4 and made algorithmic in Section 5. We believe that this decomposition result can be combined with Grohe's dynamic programming techniques for such decompositions [Gro03] to obtain PTASs for many graph problems that can be solved in bounded-treewidth graphs and bipartite graphs with a bounded number of apices, such as minimum vertex cover, maximum clique, and maximum independent set. In addition, we believe that the decomposition result can be combined with the sublinear parameter-treewidth bounds and complex dynamic programming techniques of [DFHT05] to obtain subexponential fixed-parameter algorithms for such problems, with running times $2^{O(\sqrt{k})} n^{O(1)}$.

Our proof of Theorem 1.2 uses recent deep results concerning odd S-paths, which generalize the well-known Mader's S-paths theorem, and odd clique minors. In addition, we use the seminal structure theorem in Graph Minors.

Roughly, our proof proceeds as follows. If G does not contain a huge clique minor, then we just apply the Graph Minors structure theorem. So we may assume that G has a huge clique minor. We can either translate it into a huge odd clique minor or else conclude that the component containing a huge clique minor is a "nearly" bipartite graph, with the help of the recent deep results. In the second case, this component becomes one of the pieces of the decomposition, and this helps us find a decomposition.

Our decomposition Theorem 1.1 allows us to partition the graph into two parts of bounded treewidth. In contrast with the H-minor-free case [DHK05], this result is no longer true if we require more than two parts. Therefore, in this sense, odd-minor-free graphs behave very differently from H-minor-free graphs.

2 Basics of Graph Minor Decomposition

This section describes the Robertson-Seymour decomposition theorem characterizing the structure of H-minor-free graphs and the relevant basic concepts.

First we define the basic notion of treewidth, introduced by Robertson and Seymour [RS86]. To define this notion, first we consider a representation of a graph as a tree, called a tree decomposition. Precisely, a *tree decomposition* of a graph $G = (V, E)$ is a pair (T, χ) in which $T = (I, F)$ is a tree and $\chi = \{\chi_i \mid i \in I\}$ is a family of subsets of $V(G)$ such that

1. $\bigcup_{i \in I} \chi_i = V$;
2. for each edge $e = \{u, v\} \in E$, there exists an $i \in I$ such that both u and v belong to χ_i; and
3. for all $v \in V$, the set of nodes $\{i \in I \mid v \in \chi_i\}$ forms a connected subtree of T.

To distinguish between vertices of the original graph G and vertices of T in the tree decomposition, we call vertices of T *nodes* and their corresponding χ_i's *bags*. The *width* of the tree decomposition is the maximum size of a bag in χ minus 1. The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum width over all possible tree decompositions of G. A tree decomposition is called a *path decomposition* if $T = (I, F)$ is a path. The *pathwidth* of a graph G, denoted $\mathrm{pw}(G)$, is the minimum width over all possible path decompositions of G.

Second, we need a basic notion of embedding; see, e.g., [RS94, CM05]. In this paper, an *embedding* refers to a *2-cell embedding*, i.e., a drawing of the vertices and edges of the graph as points and arcs in a surface such that every face (region outlined by edges) is homeomorphic to a disk. A *noose* in such an embedding is a simple closed curve on the surface that meets the graph only at vertices. The *length* of a noose is the number of vertices it visits. The *representativity* or *face-width* of an embedded graph is the length of the shortest noose that cannot be contracted to a point on the surface.

At a high level, the deep decomposition theorem of Robertson and Seymour [RS03, Theorem 1.3] says that, for every graph H, every H-minor-free graph can be expressed as a "tree structure" of pieces, where each piece is a graph that can be drawn in a surface in which H cannot be drawn, except for a bounded number of "apex" vertices and a bounded number of "local areas of nonplanarity" called "vortices". Here the bounds depend only on H. To make this theorem precise, we need to define each of the notions in quotes.

Each piece in the decomposition is "h-almost-embeddable" in a bounded-genus surface where h is a constant depending on the excluded minor H. Roughly speaking, a graph G is *h-almost embeddable* in a surface S if there exists a set X of size at most h of vertices, called *apex vertices* or *apices*, such that $G - X$ can be obtained from a graph G_0 embedded in S by attaching at most h graphs of pathwidth at most h to G_0 within h faces in an orderly way. More precisely, a graph G is *h-almost embeddable* in S if there exists a vertex set X of size at most h (the *apices*) such that $G - X$ can be written as $G_0 \cup G_1 \cup \cdots \cup G_h$, where

1. G_0 has an embedding in S;
2. the graphs G_i, called *vortices*, are pairwise disjoint;
3. there are faces $F_1, \ldots, F_h$ of G_0 in S, and there are pairwise disjoint disks $D_1, \ldots, D_h$ in S, such that for $i = 1, \ldots, h$, $D_i \subset F_i$ and $U_i := V(G_0) \cap V(G_i) = V(G_0) \cap D_i$; and
4. the graph G_i has a path decomposition $(\mathcal{B}_u)_{u \in U_i}$ of width less than h, such that $u \in \mathcal{B}_u$ for all $u \in U_i$. The sets $\mathcal{B}_u$ are ordered by the ordering of their indices u as points along the boundary cycle of face F_i in G_0.

An h-almost embeddable graph is *apex-free* if the set X of apices is empty.

The pieces of the decomposition are combined according to "clique-sum" operations, a notion which goes back to characterizations of $K_{3,3}$-minor-free and K_5-minor-free graphs by Wagner [Wag37] and serves as an important tool in the Graph Minor Theory. Suppose G_1 and G_2 are graphs with disjoint vertex sets and let $k \geq 0$ be an integer. For $i = 1, 2$, let $W_i \subseteq V(G_i)$ form a clique of size k and let G'_i be obtained from G_i by deleting some (possibly no) edges from the induced subgraph $G_i[W_i]$ with both endpoints in W_i. Consider a bijection $h : W_1 \to W_2$. We define a *k-sum* G of G_1 and G_2, denoted by $G = G_1 \oplus_k G_2$ or simply by $G = G_1 \oplus G_2$, to be the graph obtained from the union of G'_1 and G'_2 by identifying w with $h(w)$ for all $w \in W_1$. The images of the vertices of W_1 and W_2 in $G_1 \oplus_k G_2$ form the *join set*. Note that each vertex v of G has a corresponding vertex in G_1 or G_2 or both. Also, $\oplus$ is not a well-defined operator: it can have a set of possible results.

Now we can finally state a precise form of the decomposition theorem:

THEOREM 2.1. [RS03, Theorem 1.3] *For every graph H, there exists an integer $h \geq 0$ depending only on $|V(H)|$ such that every H-minor-free graph can be obtained by at most h-sums of graphs that are h-almost-embeddable in some surfaces in which H cannot be embedded.*

In particular, if H is fixed, any surface in which H cannot be embedded has bounded genus. Thus, the summands in the theorem are h-almost-embeddable in bounded-genus surfaces.

A polynomial-time algorithm for computing the structure guaranteed by this theorem was recently obtained in [DHK05].

3 Partitioning Odd-Minor-Free Graphs into Bounded-Treewidth Graphs

In this section, we prove Theorem 1.1 about partitioning any odd-K_k-minor-free graph into two induced subgraphs of bounded treewidth. Our proof uses the decomposition result for odd-minor-free graphs from Theorem 1.2, which is proved in the rest of the paper. We also need the following result from [DHK05]:

THEOREM 3.1. [DHK05, Theorem 3.1] *For any fixed graph H, there is a constant c_H such that, for any integer $k \geq 1$ and for every H-minor-free graph G, the vertices of G (or the edges of G) can be partitioned into $k+1$ sets such that any k of the sets induce a graph of treewidth at most $c_H\, k$. Furthermore, such a partition can be found in polynomial time.*

Now we proceed to the proof of Theorem 1.1.

Proof of Theorem 1.1: By Theorem 1.2, every odd-K_k-minor-free graph can be written as a clique sum $P_1 \oplus P_2 \oplus \cdots \oplus P_\ell$ of either h-almost-embeddable graphs or bipartite graphs together with at most h_k apex vertices such that the ith clique sum $(P_1 \oplus P_2 \oplus \cdots \oplus P_i) \oplus P_{i+1}$ has join set J_{i+1} contained in the set X_{i+1} of apices in piece P_{i+1}.

We prove the statement of Theorem 1.1 by induction on i. Suppose $i = 1$. If P_i is an h-almost embeddable graph, then we are done by Theorem 3.1. Also, if P_i is a bipartite graph W together with at most h_k apex vertices, clearly we can 2-partition P_i by taking each partite set of W together with some vertices in apex vertices. Note that the labeling of apex vertices is arbitrary. It is easy to see that the treewidth of each set is at most h_k+1, and hence it is at most c_k. (Here, we assume that $c_k \geq h_k + 1$.)

Suppose by induction that $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ has a labeling with two labels such that each label induces a graph of treewidth at most c_k. We merge the labelings of $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ and P_{i+1} by preferring the former labeling for any vertex in the join set J_{i+1}. Because $J_{i+1} \subseteq X_{i+1}$, this labeling of J_{i+1} is just a particular choice for the arbitrary labeling of X_{i+1}. By [DHN[+]04, Lemma 3], for any two graphs G' and G'', $\mathrm{tw}(G' \oplus G'') \leq \max\{\mathrm{tw}(G'), \mathrm{tw}(G'')\}$. Thus, the treewidth of each label set in $(P_1 \oplus P_2 \oplus \cdots \oplus P_i) \oplus P_{i+1}$ is at most the maximum of the treewidth of each label set within $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ and the treewidth of each label set within P_{i+1}. The latter is at most c_k as argued above, and the former is at most c_k by the induction hypothesis. Therefore the label sets form the desired partition.

The construction of the label sets runs in linear time given the decomposition from Theorem 1.1, for a polynomial overall time bound. $\square$

Theorem 1.1 is the best analog of Theorem 3.1 that can be obtained for odd-minor-free graphs. The two-set partition is the best possible because the complete bipartite graph $K_{n/2,n/2}$ cannot be partitioned into $k+1 > 2$ sets such that any k of the sets induce a bounded-treewidth graph. (For example, for $k = 2$, each side of the bipartition must have at least $n/6$ vertices in a single set, and choosing these two sets and omitting the third one induces $K_{n/6,n/6}$.) The partition with respect to vertices is best possible because any partition of the edges of $K_{n/2,n/2}$ into $O(1)$ sets has $\Omega(n^2)$ edges in one set, which is impossible for a graph of bounded treewidth.

4 Structure Theorem for Odd-Minor-Free Graphs

In this section, we prove Theorem 1.2 about the structure of odd-H-minor-free graphs, which is the foundation for our paper. Later, in Section 5, we show that this structure can be computed algorithmically. This theorem generalizes the main structural result for H-minor-free graphs developed in Graph Minor Theory [RS03] (see Section 2).

Our proof is by induction on the number of vertices. For inductive purposes, we prove the following somewhat stronger statement:

THEOREM 4.1. *For any odd K_k-minor-free graph G and any vertex set Z with at most Θ vertices where Θ comes from Theorem 3.1 in Graph Minors XVI [RS03] (and hence depends only on k), G can be obtained by clique-sums (h-sums) of the following two classes of graphs:*

1. *bipartite graphs together with a bounded number of apex vertices (at most $|\Theta| + 16k$), and*
2. *h-almost-embeddable graphs, where h is the constant in Theorem 4.2 below for $H = K_{32k,\,(16k-1)\binom{32k}{16k}+1}$.*

In addition, Z is contained in the apex vertex set of some bag (a "bag" is one of the summands in the clique sum). Moreover if $G_1 \oplus G_2$ is a clique sum and G_2 is a child of G_1, then $G_1 \oplus G_2$ is contained in the apex vertex set of G_2. Furthermore, if G_1 is a bipartite graph W together with bounded number of apex vertices, say at most h_k apex vertices, then $|G_2 \cap W| \leq 1$.

This theorem is in contrast with the remarkable following theorem in [RS03].

THEOREM 4.2. [RS03, Theorem 1.3] *For every graph H, there exists an integer $h \geq 0$ depending only on $|V(H)|$ such that every H-minor-free graph can be obtained by at most h-sums of graphs that are h-almost-embeddable in some surfaces in which H cannot be embedded.*

We need the following result, proved in [KS]. For completeness, we include a proof.

THEOREM 4.3. *Suppose G has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor. Then either G has an odd-K_k-minor or G has a vertex set X of order at most $8k$ such that $G-X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G-X$ that do not intersect F or blocks with a cut vertex to F. Furthermore, F hits all but at most $8k$ nodes of the original $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor.*

Proof: A complete minor of order l can be thought of l vertex disjoint trees, every two of which are joined by an edge. We call such a minor *even* if the union of these trees is bipartite. We call such a minor *odd* if its vertices can be two-colored so that the edges in the trees are bichromatic but the edges between two disjoint trees are monochromatic.

Geelen et al. [GGG$^+$04] proved the following result.

THEOREM 4.4. *If G has an even complete minor of order at least $16k$, then either G has an odd complete minor of order k or G has a vertex set X with $|X| < 8k$ such that $G - X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. Furthermore, F hits all but at most $8k$ nodes of the original even complete minor of order at least $16k$.*

Now we can think of a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor as follows:

> There are $32k + (16k - 1)\binom{32k}{16k} + 1$ disjoint trees $T_1, \dots, T_{32k}, T'_1, \dots, T'_{(16k-1)\binom{32k}{16k}+1}$ such that there is an edge between T_i and T'_j for any i, j with $1 \le i \le 32k$ and $1 \le j \le (16k-1)\binom{32k}{16k}+1$.

We first two-color (using colors 1 and 2) the trees $T_1, \dots, T_{32k}$ such that each T_i is bichromatic. Then for each j, we two-color T'_j in such a way that T'_j is bichromatic and there are at least $16k$ bichromatic edges between T'_j and $\bigcup_{i=1}^{32k} T_i$, for $1 \le j \le (16k - 1)\binom{32k}{16k} + 1$. This is possible because we have two choices for two-coloring of T'_j. Then by the Pigeonhole Principle, there are $16k$ disjoint trees in $\{T_1, \dots, T_{32k}\}$, say trees $T_1, \dots, T_{16k}$, and there are $16k$ disjoint trees in $\{T'_1, \dots, T'_{(16k-1)\binom{32k}{16k}+1}\}$, say trees $T'_1, \dots, T'_{16k}$, in such a way that each edge between T_i and T'_j is bichromatic for $i = 1, \dots, 16k$ and $j = 1, \dots, 16k$.

Now let $T^*_i = T_i \cup T'_i$, where $i = 1, \dots, 16k$. Clearly $\bigcup_{i=1}^{16k} T^*_i$ is bipartite and forms an even complete minor of order $16k$ in G. By Theorem 4.4, either G has an odd complete minor of order k or G has a vertex set X of order at most $8k$ such that $G - X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. Furthermore, F hits all the nodes of the original even complete minor of order $16k$. Because the size of X is at most $8k$, and $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$ has minimum degree at least $32k$, it is easy to see that F hits all the nodes of the original minor of $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$.

This completes the proof of Theorem 4.3. □

Let us remark that in the proof of this result, once we detect a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor, we can find an even complete minor of order $16k$ in the polynomial time, actually, in the linear time. By the result of Robertson and Seymour [RS95], we can actually detect the minor in $O(n^3)$. Hence we can detect the even minor, too. It remains to find a polynomial-time algorithm for Theorem 4.4. The proof of Theorem 4.4 in [GGG$^+$04] certainly implies the polynomial-time algorithm to find the desired conclusion. Actually, it detects either a desired odd minor or a vertex set X of bounded number of vertices in G such that $G - X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. Furthermore, F hits all but at most $8k$ nodes of the original even complete minor of order at least $16k$. Geelen et al. [GGG$^+$04] reduces this problem to the problem of finding the maximum matching which can be solved in $O(n^3)$ time; see [Gab73, Law76, CM78, GMG82].

Now we are ready to prove Theorem 4.1.

Proof of Theorem 4.1: Suppose G has a separation (A, B) of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty. Furthermore, suppose that both $|(Z \cap A) \cup (A \cap B)| \le \Theta$ and $|(Z \cap B) \cup (A \cap B)| \le \Theta$. We first apply induction to A with $Z = (Z \cap A) \cup (A \cap B)$. We also apply induction to B with $Z = (Z \cap B) \cup (A \cap B)$. If we glue A and B at $A \cap B$, then it is easy to see that the resulting decomposition is as desired in Theorem 4.1 because $A \cap B$ is contained in one bag of the decompositions of A and B, respectively. Actually, $A \cap B$ is contained in the apex vertex set of one bag of the decompositions of A and B, respectively. Also both $Z \cap A$ and $Z \cap B$ are contained in the apex vertex sets of bags of the decompositions of A and B, respectively. So we can glue the decompositions of A and B at $A \cap B$. Hence for any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty, we may assume that either $|(Z \cap A) \cup (A \cap B)| > \Theta$ or $|(Z \cap B) \cup (A \cap B)| > \Theta$.

Then this defines tangle T of order $\Theta/2$, assuming that $|(Z \cap B) \cup (A \cap B)| > \Theta$ for any separation (A, B) of order at most $\Theta/2$. This "tangle" will not be used in the proof, but we mention it out of interest.

From now on, we assume that $\Theta > 16k$. So $\Theta/2 > 8k$, which we shall use many times in our proof.

Suppose G has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M such that there is no separation (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z, and B hits all the nodes in M. Then by Theorem 4.3, G has a vertex set X of order at most $8k$ such that $G - X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. Furthermore, F hits all but at most $8k$ nodes of the original minor M. Moreover, each component in $G - X - F$ has at most $\Theta/2$ vertices of Z by our assumption.

Let us now observe that there is one big component W

in $G-X$ such that W contains a bipartite subgraph F and each odd cycle is contained in either components of $G-X$ that do not intersect F or blocks in W with a cut vertex to F. For any block or any component, say B, in $G-X$ such that $|B \cap Z| + |X|$ is at most $\Theta - 1$, we apply the induction with $G = B \cup X$ and $Z = X \cup \{v\} \cup (B \cap Z)$, where v is a cut vertex of $G-X$ if B is a block. By our assumption, there is no block or component B such that $|B \cap Z| + |X|$ is at least Θ. Hence the induction hypothesis is satisfied for $B \cup X$ for each component or block B. Then we get a desired decomposition of $B \cup X$ for each block and each component B of $G-X$ such that all the vertices in Z are in the apex vertex set of one bag, and if $G_1 \cap G_2$ is a clique-sum, where G_2 is a child of G_1 and G_1 consists of a bipartite graph W with at most h_k apex vertices, then $|W \cap G_2| \leq 1$. In addition, if $G_1 \cap G_2$ is a clique-sum, where G_2 is a child of G_1, then $G_1 \cap G_2$ is contained in the apex vertex set of G_2. In addition, we can glue all these decompositions at $Z \cup X \cup \{v\}$, where v is a cut vertex for the corresponding block, because each decomposition has a bag such that Z is contained in the apex vertex set of the bag. Let us observe that $F - Z$ together with $Z \cup X$ as apex vertices satisfies the first graph as described in Theorem 4.1. So this becomes a bag, and in fact the "root" of the resulting decomposition. Hence this resulting decomposition satisfies Theorem 4.1. So we finish the case when there is a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M such that there is no separation (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z, and B hits all the nodes in M.

If G does not contain a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor, then we can just apply Theorem 5.1 to G, and get a desired decomposition after putting all the vertices in Z in the apex vertex set of some bag.

So it remains to consider the case that G has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M, but there is a separation (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z, and B hits all the nodes in M. To prove this case, we need some theorems.

Given a subset $S \subseteq V(G)$, an *S-cut* is a pair (A, B) of nonempty subsets of $V(G)$ such that $V(G) = A \cup B$, $S \subseteq A$, $B - A \neq \emptyset$, and G has no edge joining $A - B$ to $B - A$. The order of the S-cut is $|A \cap B|$.

The next lemma, proved in [Kaw04, Kawc], is heavily used in our proof. For completeness, we include a proof inspired by the proof of Robertson and Seymour [RS95].

LEMMA 4.1. *Let G be a graph and $S = \{s_1, \ldots, s_k\}$ be a set of k vertices. Suppose G has a K_{2k}-minor and no S-cut of order less than k. Then G has vertex disjoint nonempty connected subgraphs $C_1, \ldots, C_k$ such that, each $1 \leq i \leq k$, the subgraph C_i contains s_i and is adjacent to all the subgraphs $C_1, \ldots, C_{i-1}, C_{i+1}, C_k$.*

Proof: We will prove the following slightly stronger statement, which immediately implies Lemma 4.1:

> (*) Let G be a graph and $S = \{s_1, \ldots, s_k\}$ be a set of k vertices. Suppose G contains $2k$ vertex disjoint nonempty subgraphs D_i for $1 \leq i \leq 2k$ such that each D_i is either connected or each of its components meets S, and moreover each D_i is adjacent to all D_j $(i \neq j)$ which do not meet S. Also suppose G has no S-cut (A, B) of order less than k with at least one D_i in $B - A$. Then G satisfies the conclusion of (3.1).

We prove by induction on $|V(G)|$. It is easy to check that the statement (*) is true for $|V(G)| = 2k$. Let G be a minimal counterexample to (*), that is, take G such that $|V(G)| + |E(G)|$ is as small as possible. Let $E(S)$ be edges joining two vertices in S. If all D_i's disjoint from S are single vertices, then by Hall's Theorem, there is a perfect matching between S and $G - S$, and the result easily follows because $|G - S| \geq k$. We claim that there is no D_i disjoint from S containing an edge. For suppose $e \in E(G) - E(S)$ is an edge contained in some D_i. If we contract e, then the resulting graph is either no longer counterexample or has an S-cut of order exactly k. In the former case, we are done. So, we may assume that there exists an S-cut (A, B) of order exactly k containing e. Because each D_i is adjacent to all D_j's $(i \neq j)$ which do not meet S, D_i is adjacent to at least k of D_j's. Hence for any i, D_i cannot be contained in $A - B$. Let $S' = A \cap B$, $G' = \langle B \rangle$, and let $D'_i = D_i \cap G'$ for $1 \leq i \leq 2k$. Note that $S \subseteq A$. If $S' = S$, then $G - e$ would give a minimal counterexample. So $A - B \neq \emptyset$. By the assumption and Menger's theorem, there exist k disjoint paths from S to S'. Then G', S' and D'_i for $1 \leq i \leq 2k$ satisfy the assumption of (*). Hence G' satisfies the conclusion of (*) by the induction, and so does G, a contradiction. Therefore there are no such edges. This implies that each D_i either has $|D_i| = 1$ or contains a vertex in S and has no edges except for $E(S)$. Moreover, at least k of D_i's consist of only one vertex which is not in S. Let R be the set of D_i which is not in S. We claim that there exists a matching from S to R. Otherwise, there would be an S-cut of order less than k by Hall's Theorem. Now contracting each edge of this matching would satisfy the conclusion of (*). This completes the proof of (*). $\square$

Because a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor contains a K_{24k}-minor, by Lemma 4.1, for each minimal separation (A, B) such that B hits all the nodes of a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor, we can contract B onto $A \cap B$ such that $A \cap B$ becomes a clique. We call this operation *clique reduction*.

So if the current graph G has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M, but there is a separation (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z, and B hits all the nodes in M, then we take a minimal separation (A, B), and we contract B onto $A \cap B$ such that $A \cap B$ becomes a clique. Let G' be the resulting graph. Let $Z' \subseteq ((A \cap Z) \cup (A \cap B))$, where $|Z'| \leq \Theta$ and $(A \cap Z) \subseteq Z'$. Because A contains all but at most $\Theta/2$ vertices in Z, it is easy to see that such a vertex set Z' does exist. Then we apply the whole argument above to G' with Z' as long as we can. Then the resulting graph G'' either contains

1. a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M such that there is no separation of (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z', and B hits all the nodes in M in G, or

2. no $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M.

In the first case, we claim we are done. First we apply the whole argument above to G'' (actually A). Then we can extend this decomposition of A to the whole graph G, because each time we perform the clique reduction—say there is a separation (A, B), and we shall contract $B - A$ onto $A \cap B$—the resulting graph in $A \cap B$ becomes a clique. More precisely, this clique is contained in one bag of the desired decomposition of A. Also $Z \cap A$ is contained in one bag of the desired decomposition of A. In fact, $Z \cap A$ is contained in the apex vertex set of the bag of the desired decomposition of A, and the clique $A \cap B$ can be contained only in either the torso of h-almost embeddable graphs or bipartite graphs together with at most h vertices. Then we shall extend this decomposition of A by applying the induction to B with $Z = (B \cap Z) \cup (A \cap B)$. Because A contains all but at most $\Theta/2$ vertices in Z, $|(B \cap Z) \cup (A \cap B)| \leq \Theta$. So the induction hypothesis is satisfied for B with $Z = (B \cap Z) \cup (A \cap B)$. Hence we can extend the decomposition of A to B, and if we glue A and B at $A \cap B$, and put all the vertices in $B \cap Z$ to the apex vertex set of the bag of the decomposition of A in such a way that this bag contains all the vertices of Z in the apex vertex set, then clearly this is a desired decomposition of Theorem 4.1. Note that $Z \cap A$ is contained in the apex vertex set of that bag of the decomposition of A, and $A \cap B$ is in the apex vertex set of one bag of the decomposition of B.

In the second case, we can just apply Theorem 4.2 to G''. Suppose that there is a separation (A, B), and we contracted $B - A$ onto $A \cap B$. Suppose furthermore that the resulting A satisfies the second case. Then we shall extend the decomposition by applying the induction to B with $Z = (B \cap Z) \cup (A \cap B)$. Because A contains all but at most $\Theta/2$ vertices in Z, $|(B \cap Z) \cup (A \cap B)| \leq \Theta$. So the induction hypothesis is satisfied for B with $Z = (B \cap Z) \cup (A \cap B)$. Note that each time we perform the clique reduction—say there is a separation (A, B), and we shall contract $B - A$ onto $A \cap B$—the resulting graph in $A \cap B$ becomes a clique. So this clique is contained in one bag of the desired decomposition of A. Also $Z \cap A$ is contained in one bag of the desired decomposition of A. In fact, $Z \cap A$ is contained in the apex vertex set of the bag of the desired decomposition of A, and the clique $A \cap B$ can be contained only in either the torso of h-almost embeddable graphs or bipartite graphs together with at most h vertices. Hence, by the same argument in the previous paragraph, we can extend the decomposition of A to B, and if we glue A and B at $A \cap B$, and put all the vertices in $B \cap Z$ to the apex vertex set of the bag of the decomposition of A in such a way that this bag contains all the vertices of Z in the apex vertex set, then clearly this is a desired decomposition of Theorem 4.1. Note that $Z \cap A$ is contained in the apex vertex set of that bag of the decomposition of A, and $A \cap B$ is in the apex vertex set of one bag of the decomposition of B.

This completes the proof. $\square$

Section 6 gives an alternate proof of Theorem 4.1. While this alternate proof seems shorter, it is difficult if not impossible to make algorithmic without reworking through a major part of Graph Minor Theory. See the discussion in that section.

5 Algorithm for Structure of Odd-Minor-Free Graphs

In this section, we give a polynomial-time algorithm to compute the structure guaranteed by Theorem 4.1. First we need the following theorem from [DHK05].

THEOREM 5.1. [DHK05] *There is a polynomial-time algorithm to obtain a decomposition as described in Theorem 4.2 for H-minor-free graphs. Actually, we can specify a vertex set Z with $|Z| \leq \Theta$ so that Z is contained in the apex vertex set of the desired decomposition as described in Theorem 4.2.*

Now we are ready to describe our algorithm for Theorem 4.1. This algorithm is based on the proof of Theorem 4.1 from Section 4.

Algorithm for Theorem 4.1

Input: A graph G and $Z \subseteq V(G)$ with $|Z| \leq \Theta$.

Output: As described in Theorem 4.1.

Running time: $n^{O(h)}$.

Description:

Step 1. Test whether G has a separation (A, B) of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty. Suppose that both $|(Z \cap A) \cup (A \cap B)| \leq \Theta$ and $|(Z \cap B) \cup (A \cap B)| \leq \Theta$. Then we first apply this algorithm to A with $Z = (Z \cap A) \cup (A \cap B)$, recursively. Then we apply this algorithm to B with $Z = (Z \cap B) \cup (A \cap B)$, recursively. Then we glue A and B at $A \cap B$. Then it is easy to see that the resulting decomposition is as desired in Theorem 4.1

because $A \cap B$ is contained in one bag of the decompositions of A and B, respectively. Actually, $A \cap B$ is contained in the apex vertex set of the bag of the decomposition of A and B, respectively. Also both $Z \cap A$ and $Z \cap B$ are contained in the apex vertex set of the bags of the decomposition of A and B, respectively. So we can glue the decompositions of A and B at $A \cap B$.

If for any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ in the current graph are nonempty, either $|(Z \cap A) \cup (A \cap B)| > \Theta$ or $|(Z \cap B) \cup (A \cap B)| > \Theta$, then go to Step 2.

Step 2. From here, any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ in the current graph are nonempty, either $|(Z \cap A) \cup (A \cap B)| > \Theta$ or $|(Z \cap B) \cup (A \cap B)| > \Theta$. Test whether G' has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M. If it has, then go to Step 3. Otherwise, go to Step 7. This test can be done by the result of Robertson and Seymour [RS95]. The time complexity is $O(n^3)$. In fact, we can detect the minor in $O(n^3)$ time.

Step 3. Check whether there is a separation (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z, and B hits all the nodes in M. If there is no such a separation, then go to Step 4. Otherwise, take a minimal such separation (A, B), and delete $B - A$ and make $A \cap B$ a clique. In addition, set $Z' \subseteq ((A \cap Z) \cup (A \cap B))$, where $|Z'| \leq \Theta$ and $(A \cap Z) \subseteq Z'$. Because A contains all but at most $\Theta/2$ vertices in Z, it is easy to see that such a vertex set Z' does exist. Then go to Step 1 with the current graph.

Step 4. Find an even K_{16k}-minor by using the argument in the proof of Theorem 4.3. This can be done in polynomial time, actually in linear time if we can detect a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor in Step 2.

Step 5. Detect a separation X of order $|X| < 8k$ as described in Theorem 4.3. The proof in Geelen et al. [GGG+04] reduces this problem to the problem of finding the maximum matching that can be solved in $O(n^3)$ time; see [Gab73, Law76, CM78, GMG82]. So it takes at most $O(n^3)$ time.

Step 6. We have one big component W in $G - X$ such that W contains a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. For any block or any component, say B, in $G - X$ such that $|B \cap Z| + |X|$ is at most $\Theta - 1$, we apply this algorithm recursively with $Z = X \cup \{v\} \cup (B \cap Z)$, where v is a cut vertex of $G - X$ if B is a block. Note that there are no block nor component B such that $|B \cap Z| + |X|$ is at least Θ by Step 3. Now $F - Z$ together with $Z \cup X$ becomes one of the bag, and each block and each component of $G - X - F$ becomes a desired decomposition such that all the vertices in Z are in the apex vertex set of some bag. In addition, we can glue all these decompositions at $Z \cup X \cup \{v\}$, where v is a cut vertex for the corresponding block, because each decomposition has a bag such that Z is contained in the apex vertex set of the bag. Hence the resulting decomposition satisfies Theorem 4.1.

Step 7. At this moment, G does not have a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor. So we just apply the algorithm of Theorem 5.1 to G, and output the resulting decomposition.

Step 8. Finally, we shall glue two graphs, and repeat this recursively. More precisely, suppose $G' = G_1 \cup G_2$ and $Z' = G_1 \cap G_2$. If both G_1 and G_2 have a desired decomposition such that $Z' \in Z_1$ and $Z' \in Z_2$, where Z_1 is Z for G_1 in Theorem 4.1 and Z_2 is Z for G_2 in Theorem 4.1, and in addition, both Z_1 and Z_2 are contained in the apex vertex set of one bag of the decomposition of G_1 and G_2, respectively, then we glue G_1 and G_2 with Z'. We repeat this procedure recursively until the end. Also, if there is a separation (A, B) as in Step 3, then we can extend the decomposition of A to the whole graph G. Note that $A \cap B$ becomes a clique. So this clique is contained in one bag. Also $Z \cap A$ is contained in one bag of the desired decomposition of A. In fact, $Z \cap A$ is contained in the apex vertex set of the bag of the desired decomposition of A, and the clique $A \cap B$ can be contained only in either the torso of h-almost embeddable graphs or bipartite graphs together with at most h vertices. Then we extend the decomposition of A by applying this algorithm to B with $Z = (B \cap Z) \cup (A \cap B)$. Because A contains all but at most $\Theta/2$ vertices in Z, $|(B \cap Z) \cup (A \cap B)| \leq \Theta$. So we can apply this algorithm to B with $Z = (B \cap Z) \cup (A \cap B)$. Hence we can extend the decomposition of A to B, and if we glue A and B at $A \cap B$, and put all the vertices in $B \cap Z$ into the apex vertex set of the bag of the decomposition of A in such a way that this bag contains all the vertices of Z in the apex vertex set, then clearly this is a desired decomposition. Note that $Z \cap A$ is contained in the apex vertex set of that bag of the decomposition of A, and $A \cap B$ is in the apex vertex set of one bag of the decomposition of B. This completes the description of the algorithm.

The correctness of the algorithm follows from the proof of Theorem 4.1, but for the completeness, we shall give some remarks, and sketch the proof.

This algorithm is constructive, in particular, in Step 6, we can get one bag that consists of bipartite graphs with at most $\Theta + 16k$ apex vertices. Furthermore, once we conclude that the current graph does not contain a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor, then the algorithm uses Theorem 5.1. Moreover, at Step 1, suppose G has a separation (A, B) of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty. Furthermore, suppose both $|(Z \cap A) \cup (A \cap B)| \leq \Theta$ and $|(Z \cap B) \cup (A \cap B)| \leq \Theta$. Then we first apply the algorithm to A with $Z = (Z \cap A) \cup (A \cap B)$. Then we also apply the algorithm to B with $Z = (Z \cap B) \cup (A \cap B)$. If we glue A and B at $A \cap B$, then it is easy to see that the resulting decomposition is as desired in Theorem 4.1 because $A \cap B$ is contained in one bag of the decompositions of A and B, respectively. Actually, $A \cap B$ is contained in the apex vertex set of the bag of the decompositions of A and B, respectively. Also, both $Z \cap A$ and $Z \cap B$ are contained in the apex vertex set of the bag of the decompositions of A and B, respectively. So we can glue the decompositions of A and B at $A \cap B$. Hence after Step 1, for any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty, either $|(Z \cap A) \cup (A \cap B)| > \Theta$ or $|(Z \cap B) \cup (A \cap B)| > \Theta$.

We may now assume that there is a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor.

In Step 3, if there is a separation (A, B) of order at most $8k$ in such a way that A contains all but at most $\Theta/2$ vertices in Z, and B hits all but at most $8k$ nodes in the minor M, then as we did in the previous section, take a minimal such separation (A, B) and delete $B - A$ and make $A \cap B$ a clique. This is possible by the clique reduction as we argued in the previous section. In addition, set $Z' \subseteq ((A \cap Z) \cup (A \cap B))$, where $|Z'| \leq \Theta$ and $(A \cap Z) \subseteq Z'$. Because A contains all but at most $\Theta/2$ vertices in Z, it is easy to see that such a vertex set Z' does exist. Then by applying the algorithm repeatedly, A has a desired decomposition with the set Z'. Then we can extend this decomposition of A to the whole graph G, because each time we perform the clique reduction, the resulting graph in $A \cap B$ becomes a clique. More precisely, this clique is contained in one bag of the desired decomposition of A. Also $Z \cap A$ is contained in one bag of the desired decomposition of A. In fact, $Z \cap A$ is contained in the apex vertex set of the bag of the desired decomposition of A, and the clique $A \cap B$ can be contained only in either the torso of h-almost embeddable graphs or bipartite graphs together with at most h vertices. Then we extend the decomposition by applying the algorithm to B with $Z = (B \cap Z) \cup (A \cap B)$. Because A contains all but at most $\Theta/2$ vertices in Z, $|Z| = |(B \cap Z) \cup (A \cap B)| \leq \Theta$. So the hypothesis for the algorithm to B with $Z = (B \cap Z) \cup (A \cap B)$ is satisfied. Hence we can extend the decomposition of A to B, and if we glue A and B at $A \cap B$, and put all the vertices in $B \cap Z$ to the apex vertex set of the bag of the decomposition of A in such a way that this bag contains all the vertices of Z in the apex vertex set, then clearly this is a desired decomposition of Theorem 4.1. Note that $Z \cap A$ is contained in the apex vertex set of that bag of the decomposition of A, and $A \cap B$ is in the apex vertex set of one bag of the decomposition of B.

Also, by Theorem 4.3, because we know that G has no odd-K_k-minors, it must contain a separation X as described in Theorem 4.3. Hence, in each block and component B of $G - X$ such that $|B \cap Z| + |X| \leq \Theta$, we can apply this algorithm recursively with $Z = X \cup \{v\} \cup (B \cap Z)$, where v is a cut vertex of $G - X$ if B is a block, to $B \cup X$. Note that by Step 3, for each component or block B, $|B \cap Z| + |X| < \Theta$. Hence the hypothesis for the algorithm on $B \cup X$ is satisfied for each component or block B. Then we get a desired decomposition for each block and each component of $G - X$ such that all the vertices in Z are in the apex vertices of one bag, and if $G_1 \cap G_2$ is a clique-sum, where G_2 is a child of G_1 and G_1 consists of a bipartite graph W with at most h_k apex vertices. Moreover $|W \cap G_2| \leq 1$. In addition, if $G_1 \cap G_2$ is a clique-sum, where G_2 is a child of G_1, then $G_1 \cap G_2$ is contained in the apex vertex set of G_2. In addition, we can glue all these decompositions at $Z \cup X \cup \{v\}$, where v is a cut vertex for the corresponding block, because each decomposition has a bag such that Z is contained in the apex vertex set of the bag. Let us observe that $F - Z$ together with $Z \cup X$ as apex vertices satisfies the first outcome as described in Theorem 4.1. So this becomes a bag, and in fact the "root" of the resulting decomposition. Hence this resulting decomposition satisfies Theorem 4.1.

Finally, let us estimate the time complexity of the algorithm. We need to detect the minor of $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$ in Step 2. This takes $O(n^3)$ time by [RS95]. In Step 2, we need to detect the separation (A, B). This can be done by the algorithm of Henzinger, Rao, and Gabow [HRG00] which needs $O(n^2)$ time. Another n pops up because we may use this step recursively. Also it takes $O(n^3)$ time to detect X in Step 4, as we remarked just after the proof of Theorem 4.3. So, in Step 5, we run $O(n^4)$ times. In Step 7, because we run Theorem 5.1, it takes $n^{o(h)}$. Hence this is the most expensive part.

This completes the analysis of the correctness and of the stated time complexity of the algorithm.

Let us observe that we can detect the odd K_k-minor provided that G has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor. To see this, we first detect a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor by Robertson and Seymour [RS95]. Then the argument in the proof of Theorem 4.3 allows us to detect the desired odd-minor, as we remarked just after the proof of Theorem 4.3. As we noted before, the proof of Theorem 4.4 in [GGG$^+$04] certainly implies a polynomial-time algorithm to find the desired conclusion of Theorem 4.4. Actually, it detects either a desired odd minor or a vertex set X of a bounded number of vertices in G such that $G - X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. Moreover, B hits all but at most $8k$ nodes in the minor. The time complexity is $O(n^3)$. Hence we can detect the desired odd-minor if the outcome (2) of Theorem 4.4 holds, provided that there is a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor. □

6 Alternate Proof of Theorem 4.1

Here we give an alternate proof of Theorem 4.1. This proof is easier and shorter for obtaining the existential result, but it is difficult if not impossible to make algorithmic without reworking through a major part of Graph Minor Theory. See the discussion at the end of the proof.

We follow the notation in the first proof of Theorem 4.1. The easiest way to prove this theorem is to start with the grid minor controlled by this tangle T of order $\Theta/2$. We assume that the order of this tangle T is big enough to apply Theorem 3.1 in Graph Minors XVI. So Θ is as in Theorem 3.1 in Graph Minor XVI. We apply Theorem 3.1 in Graph Minors XVI with the minor $L = K_{32k,\,(16k-1)\binom{32k}{16k}+1}$ and this tangle T to G. We may assume that G contains a minor L controlled by the tangle T; otherwise, it follows from [RS03, Theorem 3.1] that we can get a desired decomposition. So assume that there is a minor L in G controlled by the tangle T. If L is controlled by this tangle T, then we know that there is no separation (A, B) of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty, and furthermore, both $|(Z \cap A) \cup (A \cap B)| \leq \Theta$ and $|(Z \cap B) \cup (A \cap B)| \leq \Theta$. In addition, there is no separation of order at most $8k$ such that both $B - A$ and $A - B$ are nonempty, and furthermore, $|(Z \cap B) \cup (A \cap B)| \leq \Theta$ and $B - A$ strictly contains a node of L. (This also follows from the fact that the starting point of the proof of Theorem 3.1 in Graph Minor XVI is the grid minor controlled by this tangle T of order $\Theta/2$, and for any separation $(A, B) \in T$, $B - A$ contains most of the vertices in this grid, i.e., $B - A$ contains all but at most $|A \cap B|$ vertices of this grid.) Let us recall that we say that the tangle T *controls* the minor L if, for any separation $(A, B) \subseteq T$ of order at most $8k$ in G, at most $8k - 1$ nodes of L is strictly contained in $A - B$.

Because G does not contain an odd K_k-minor, by Theorem 4.3, G has a vertex set X of order at most $8k$ such that $G - X$ has a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. Furthermore, F hits all but at most $8k$ nodes of the original minor L, so there is no separation (A, B) of order at most $8k$ such that both $A - B$ and $B - A$ strictly contains a node of L. Moreover, there is no block or component B such that $|B \cap Z| + |X|$ is at least Θ because L is controlled by the above tangle T and F hits all but at most $8k$ nodes of the original minor L. This means the following. $G - X$ consists of the bipartite graph F together with blocks $B_1, \ldots, B_l$ for some l and components $C_1, \ldots, C_p$ for some p such that each block B_i has an odd cycle for all i, each block B_i contains at most $\Theta/2$ vertices of Z, and each component C_j contains at most $\Theta/2$ vertices of Z. Then for each block B_i with $1 \leq i \leq l$, we apply induction with $Z = (B_i \cap Z) \cup X \cup \{v\}$, where $v = B_i \cap F$, to $B_i \cup Z$. Note that $|(B_i \cap Z) \cup X \cup \{v\}| \leq \Theta$. So the induction hypothesis is satisfied for $B_i \cup Z$ for each block B_i. Furthermore, for each component C_i with $1 \leq i \leq p$, we apply induction with $Z = (C_i \cap Z) \cup X$ to $C_i \cup X$. Note that $|(C_i \cap Z) \cup X| \leq \Theta$. So the induction hypothesis is satisfied for $C_i \cup X$ for each component C_i. Hence we get a desired decomposition for each block and each component of $G - X - F$ such that all the vertices in Z are in the apex vertices of one bag, and if $G_1 \oplus G_2$ is a clique-sum, where G_2 is a child of G_1 and G_1 consists of a bipartite graph W with at most h_k apex vertices, then $|W \cap G_2| \leq 1$. In addition, $G_1 \oplus G_2$ is contained in the apex vertex set of G_2. In addition, we can glue all these decompositions at $Z \cup X \cup \{v\}$, where v is a cut vertex for the corresponding block, because each decomposition has a bag such that Z is contained in the apex vertex set of the bag. Let us observe that $F - Z$ together with $Z \cup X$ as apex vertices satisfies the first outcome as described in Theorem 4.1. So this becomes a bag, and in fact the "root" of the resulting decomposition. Hence this resulting decomposition satisfies Theorem 4.1. This completes the alternate proof of Theorem 4.1.

As we see here, the difference between the first proof and the second proof is that, if we start with the tangle T, and detect the minor $L = K_{32k,\,(16k-1)\binom{32k}{16k}+1}$ controlled by this tangle T, then the proof is much shorter and easier. But on the other hand, this approach has some problems, which could be resolved by following the whole series of Graph Minors papers or [DHK05]. The biggest problem is that the algorithm in Theorem 5.1 assumes that G excludes L as a minor. This assumption makes the proof much simpler than the whole Graph Minors argument, simply because we may assume that we can detect the apex vertex set, so the resulting structure is as described in Theorem 4.2. But once we do not confirm whether a given graph G has L as a minor controlled by the tangle T, then the situation becomes much more difficult. Let us observe that we can test whether G has L as a minor by the algorithm of Robertson and Seymour [RS95]. But for the shorter proof, we need to test whether G has L as a minor controlled by the tangle T. To do this, we need to follow the whole Graph Minors argument, and need to rework the argument in [DHK05], making sure that the result in [DHK05] is still valid if we replace "no L-minor" by "no minor L controlled by the tangle T". We believe that this could be done, with a lot of additional work, leading to the following claim:

CLAIM 6.1. *Let T be a tangle of order Θ. There is a polynomial-time algorithm to obtain either an H minor controlled by the tangle T or a decomposition as described in Theorem 4.2 for H-minor-free graphs. Actually, in the second case, we can specify a vertex set Z with $|Z| \leq \Theta$ so that Z is contained in the apex vertex set of the desired decomposition as described in Theorem 4.2.*

However, again, the correctness of Claim 6.1 would need the whole argument of the Graph Minors papers, and reworking and extending the long arguments in [DHK05]. For this reason, the main body of the paper uses the lengthier proof of Theorem 4.1 and the algorithm based on that proof, which is self-contained and has no such dependency.

If we assume, though, that we have Claim 6.1, then we can give an algorithm for Theorem 4.1 based on the shorter proof above:

Algorithm for Theorem 4.1, Version 2.

Input: A graph G and $Z \subseteq V(G)$ with $|Z| \leq \Theta$.

Output: As described in Theorem 4.1.

Running time: $n^{O(h)}$.

Description:

Step 1. Test whether G has a separation (A, B) of order at most $\Theta/2$ such that both $B - A$ and $A - B$ are nonempty. Suppose that both $|(Z \cap A) \cup (A \cap B)| \leq \Theta$ and $|(Z \cap B) \cup (A \cap B)| \leq \Theta$. Then we first apply this algorithm to A with $Z = (Z \cap A) \cup (A \cap B)$, recursively. Then we apply this algorithm to B with $Z = (Z \cap B) \cup (A \cap B)$, recursively. Then we glue A and B at $A \cap B$. Then it is easy to see that the resulting decomposition is as desired in Theorem 4.1 because $A \cap B$ is contained in one bag of the decompositions of A and B, respectively. Actually, $A \cap B$ is contained in the apex vertex set of the bag of the desired decomposition of A and B, respectively. Also both $Z \cap A$ and $Z \cap B$ are contained in the apex vertex set of the bags of the decomposition of A and B, respectively. So we can glue the decompositions of A and B at $A \cap B$.

If for any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ in the current graph are nonempty, either $|(Z \cap A) \cup (A \cap B)| > \Theta$ or $|(Z \cap B) \cup (A \cap B)| > \Theta$, then go to Step 2.

Step 2. From here, any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ in the current graph are nonempty, either $|(Z \cap A) \cup (A \cap B)| > \Theta$ or $|(Z \cap B) \cup (A \cap B)| > \Theta$. This defines the tangle T of order at least $\Theta/2$ assuming that any separation of order at most $\Theta/2$ such that both $B - A$ and $A - B$ in the current graph are nonempty, $|(Z \cap B) \cup (A \cap B)| > \Theta$. So, now we detect all the separations (A, B) of order at most $\Theta/2$ which are created by this tangle T. Test whether G' has a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor M controlled by this tangle T. If it has, then go to Step 3. Otherwise, go to Step 6. This can be done by Claim 6.1. We know that once the tangle T is detected, we can find a grid minor as required in the algorithm of Theorem 4.2 by the standard flow method and the proof in [DJGT99]. Also, detecting all the separations consisting of the tangle T can be done by the standard flow method because Θ is fixed.

Step 3. Find an even K_{16k}-minor by using the argument in the proof of Theorem 4.3. This can be done in polynomial time, actually in linear time if we can detect a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor in Step 2.

Step 4. Detect a separation X of order $|X| < 8k$ as described in Theorem 4.3. The proof in Geelen et al [GGG+04] reduces this problem to the problem of finding the maximum matching that can be solved in $O(n^3)$ time; see [Gab73, Law76, CM78, GMG82]. So it takes at most $O(n^3)$ time.

Step 5. We have one big component W in $G - X$ such that W contains a bipartite subgraph F and each odd cycle is contained in either components of $G - X$ that do not intersect F or blocks with a cut vertex to F. For any block or any component, say B, in $G - X$, we apply this algorithm recursively with $Z = X \cup \{v\} \cup (B \cap Z)$, where v is a cut vertex of $G - X$ if B is a block, to $B \cup X$. Note that for each component or block B, $|B \cap Z| + |X|$ is at most $\Theta - 1$ because we start with the tangle T and L is controlled by the tangle T. Now $F - Z$ together with $Z \cup X$ becomes one of the bags, and each block and each component of $G - X - F$ becomes a desired decomposition such that all the vertices in Z are in the apex vertex set of some bag. In addition, we can glue all these decompositions at $Z \cup X \cup \{v\}$, where v is a cut vertex for the corresponding block, because each decomposition has a bag such that Z is contained in the apex vertex set of the bag. Let us observe that $F - Z$ together with $Z \cup X$ as apex vertices satisfies the first graph as described in Theorem 4.1. So this becomes a bag, and in fact the "root" of the resulting decomposition. Hence this resulting decomposition satisfies Theorem 4.1.

Step 6. At this moment, G does not have a $K_{32k,\,(16k-1)\binom{32k}{16k}+1}$-minor. So we just apply the algorithm of Theorem 5.1 to G, and output the resulting decomposition.

The time complexity and the correctness is almost same as the first version of the algorithm. So we omit them.

7 Conclusion

Our algorithms are among the first efficient algorithms for the general family of odd-minor-free graphs. Our techniques and structural theorems open the door for further development of efficient algorithms that exploit this structure. In particular, it would be interesting to generalize more of our knowledge from minor-free graphs to odd-minor-free graphs.

One of the major results of this paper is that any odd-minor-free graph can be partitioned into two induced subgraphs of bounded treewidth. Such partitions enable us to immediately approximate many NP-hard problems. The main open problem along these lines is the following. Suppose that a graph can be partitioned into k induced subgraphs each of treewidth at most w. Is there a fixed-parameter algorithm in terms of k and w that partitions the graph into $c_1\, k$ induced subgraphs each of treewidth at most $c_2\, w$, for constants c_1 and c_2? This problem generalizes the problem of fixed-parameter treewidth approximation (as in, e.g., [Ami01]), and is interesting even if c_1 and c_2 are small but nonconstant. Such algorithms would be powerful tools, even in practice, for approximating NP-hard problems in graphs with this structure for small k and w.

Acknowledgments

We thank Jim Geelen and Bruce Reed for many helpful and enlightening discussions. We also thank Paul Seymour for general insights into Graph Minor Theory.

References

[AG06] Ittai Abraham and Cyril Gavoille. Object location using path separators. In *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing*, pages 188–197, 2006.

[AH77] K. Appel and W. Haken. Every planar map is four colorable. I. Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 1977.

[AHK77] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. II. Reducibility. *Illinois Journal of Mathematics*, 21(3):491–567, 1977.

[Ami01] Eyal Amir. Efficient approximation for triangulation of minimum treewidth. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 7–15, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[APS07] Omid Amini, Stéphane Pérennes, and Ignasi Sau. Hardness and approximation of traffic grooming. In *Proceedings of 18th International Symposium on Algorithms and Computation*, December 2007. To appear.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 222–231, 2004.

[ASS07] Omid Amini, Ignasi Sau, and Saket Saurabh. Parameterized complexity of the smallest degree-constrained subgraph problem. Technical Report RR-6237, INRIA, June 2007. http://hal.inria.fr/inria-00157970/en/.

[Bak94] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery*, 41(1):153–180, 1994.

[BBCH07] Maria-Florina Balcan, Avrim Blum, T.-H. Hubert Chan, and MohammadTaghi Hajiaghayi. A theory of loss-leaders: Making money by pricing below cost. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, 2007. To appear.

[BNBH+98] Amotz Bar-Noy, Mihir Bellare, Magnús M. Halldórsson, Hadas Shachnai, and Tami Tamir. On chromatic sums and distributed resource allocation. *Information and Computation*, 140(2):183–202, February 1998.

[Bod05] Hans L. Bodlaender. Discovering treewidth. In *Proceedings of the 31st Conference on Current Trends in Theory and Practice of Computer Science*, volume 3381 of *Lecture Notes in Computer Science*, pages 1–16, Liptovský Ján, Slovakia, January 2005.

[Cat78] Paul A. Catlin. A bound on the chromatic number of a graph. *Discrete Mathematics*, 22(1):81–83, 1978.

[CGH71] Gary Chartrand, Dennis Geller, and Stephen Hedetniemi. Graphs with forbidden subgraphs. *Journal of Combinatorial Theory. Series B*, 10:12–41, 1971.

[Chl07] Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 691–701, 2007.

[CM78] W. H. Cunningham and A. B. Marsh, III. A primal algorithm for optimum matching. *Mathematical Programming Study*, (8):50–72, 1978.

[CM05] Sergio Cabello and Bojan Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. In *Proceedings of the 13th Annual European Symposium on Algorithms*, volume 3669 of *Lecture Notes in Computer Science*, pages 131–142, Palma de Mallorca, Spain, October 2005.

[DDO+04] Matt DeVos, Guoli Ding, Bogdan Oporowski, Daniel P. Sanders, Bruce Reed, Paul Seymour, and Dirk Vertigan. Excluding any graph as a minor allows a low tree-width 2-coloring. *Journal of Combinatorial Theory, Series B*, 91(1):25–41, 2004.

[DFHT05] Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005.

[DH05] Erik D. Demaine and MohammadTaghi Hajiaghayi. Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pages 682–689, Vancouver, January 2005.

[DHK05] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 637–646, Pittsburgh, PA, October 2005.

[DHM] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Dániel Marx. Minimizing movement: Fixed-parameter tractability. Submitted to STOC 2008.

[DHM07] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 278–287, New Orleans, Louisiana, January 7–9 2007.

[DHN+04] Erik D. Demaine, MohammadTaghi Hajiaghayi, Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. *Journal of Computer and System Sciences*, 69(2):166–195, September 2004.

[DJGT99] Reinhard Diestel, Tommy R. Jensen, Konstantin Yu. Gorbunov, and Carsten Thomassen. Highly connected sets and the excluded grid theorem. *Journal of Combinatorial Theory, Series B*, 75(1):61–73, 1999.

[DOSV00] Guoli Ding, Bogdan Oporowski, Daniel P. Sanders, and Dirk Vertigan. Surfaces, tree-width, clique-minors, and partitions. *Journal of Combinatorial Theory. Series B*, 79(2):221–246, 2000.

[Epp00] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3-4):275–291, 2000.

[FK98] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998.

[Gab73] H. N. Gabow. *Implementation of algorithms for maximum matching on non-bipartite graphs*. PhD thesis, Department of Computer Science, Stanford University, 1973.

[GG02] James F. Geelen and Bertrand Guenin. Packing odd circuits in Eulerian graphs. *Journal of Combinatorial Theory. Series B*, 86(2):280–295, 2002.

[GGG$^+$04] Jim Geelan, Bert Gerards, Luis Goddyn, Bruce Reed, Paul Seymour, and Adrian Vetta. The odd case of Hadwiger's conjecture. Submitted, 2004.

[GMG82] Zvi Galil, Silvio Micali, and Harold Gabow. Priority queues with variable priority and an $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 255–261, Chicago, IL, 1982.

[Gon05] Daniel Gonçalves. Edge partition of planar graphs into two outerplanar graphs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 504–512, New York, NY, USA, 2005. ACM Press.

[GP82] M. Grötschel and W. R. Pulleyblank. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters*, 1(1):23–27, 1981/82.

[Gro03] Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23(4):613–632, 2003.

[Gue01] Bertrand Guenin. A characterization of weakly bipartite graphs. *Journal of Combinatorial Theory. Series B*, 83(1):112–168, 2001.

[Gue05] Bertrand Guenin. Talk at Oberwolfach Seminar on Graph Theory, January 2005.

[Had43] H. Hadwiger. Über eine Klassifikation der Streckenkomplexe. *Vierteljschr. Naturforsch. Ges. Zürich*, 88:133–142, 1943.

[HK02] Magnús M. Halldórsson and Guy Kortsarz. Tools for multicoloring with applications to planar graphs and partial k-trees. *Journal of Algorithms*, 42(2):334–366, 2002.

[HRG00] Monika R. Henzinger, Satish Rao, and Harold N. Gabow. Computing vertex connectivity: new bounds from old techniques. *Journal of Algorithms*, 34(2):222–250, 2000.

[JT95] Tommy R. Jensen and Bjarne Toft. *Graph coloring problems*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1995. A Wiley-Interscience Publication.

[Kawa] Ken-ichi Kawarabayashi. Algorithmic aspects of the odd Hadwiger's conjecture. Submitted.

[Kawb] Ken-ichi Kawarabayashi. Minors in 7-chromatic graphs. Preprint.

[Kawc] Ken-ichi Kawarabayashi. On the connectivity of minimal and minimum counterexample to Hadwiger's conjecture. *Journal of Combinatorial Theory. Series B*. To appear.

[Kaw04] Ken-ichi Kawarabayashi. Rooted minor problems in highly connected graphs. *Discrete Mathematics*, 287(1-3):121–123, 2004.

[Ked96] Kiran S. Kedlaya. Outerplanar partitions of planar graphs. *Journal of Combinatorial Theory. Series B*, 67(2):238–248, 1996.

[Kle05] Philip N. Klein. A linear-time approximation scheme for TSP for planar weighted graphs. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 146–155, 2005.

[KM06] Ken-ichi Kawarabayashi and Bojan Mohar. Approximating chromatic number and list-chromatic number of minor-closed and odd-minor-closed classes of graphs. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, pages 401–416, 2006.

[Kos84] A. V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.

[KS] Ken-ichi Kawarabayashi and Z. Song. Some remarks on the odd Hadwiger's conjecture. *Combinatorica*. To appear.

[KT05] Ken-ichi Kawarabayashi and Bjarne Toft. Any 7-chromatic graph has K_7 or $K_{4,4}$ as a minor. *Combinatorica*, 25(3):327–353, 2005.

[Law76] Eugene L. Lawler. *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, New York, 1976.

[LR99] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[LT80] Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

[NW64] C. St. J. A. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1964.

[RS86] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

[RS94] Neil Robertson and P. D. Seymour. Graph minors. XI. Circuits on a surface. *Journal of Combinatorial Theory, Series B*, 60(1):72–106, 1994.

[RS95] Neil Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

[RS03] Neil Robertson and P. D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003.

[RSST97] Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70(1):2–44, 1997.

[RST93] Neil Robertson, Paul Seymour, and Robin Thomas. Hadwiger's conjecture for K_6-free graphs. *Combinatorica*, 13(3):279–361, 1993.

[Sch02] Alexander Schrijver. A short proof of Guenin's characterization of weakly bipartite graphs. *Journal of Combinatorial Theory. Series B*, 85(2):255–260, 2002.

[Sey77] P. D. Seymour. The matroids with the max-flow min-cut property. *Journal of Combinatorial Theory. Series B*, 23(2-3):189–222, 1977.

[Tho95] Robin Thomas. Problem Session of the 3rd Solvene Conference on Graph Theory, Bled, Slovenia, 1995.

[Tho98] Mikkel Thorup. All structured programs have small tree-width and good register allocation. *Information and Computation*, 142(2):159–181, 1998.

[Tho01] Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001.

[Tho04] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, 2004.

[Wag37] K. Wagner. Über eine Eigenschaft der eben Komplexe. *Mathematische Annalen*, 114:570–590, December 1937.

[Yan78] Mihalis Yannakakis. Node- and edge-deletion NP-complete problems. In *Conference Record of the 10th Annual ACM Symposium on Theory of Computing (San Diego, CA, 1978)*, pages 253–264, 1978.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, 2007.