Algorithmic Graph Minor Theory: Decomposition, Approximation, and Coloring

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

Abstract

At the core of the seminal Graph Minor Theory of Robertson and Seymour is a powerful structural theorem capturing the structure of graphs excluding a fixed minor. This result is used throughout graph theory and graph algorithms, but is existential. We develop a polynomial-time algorithm using topological graph theory to decompose a graph into the structure guaranteed by the theorem: a clique-sum of pieces almost-embeddable into bounded-genus surfaces. This result has many applications. In particular, we show applications to developing many approximation algorithms, including a 2-approximation to graph coloring, constant-factor approximations to treewidth and the largest grid minor, combinatorial polylogarithmic-approximation to half-integral multicommodity flow, subexponential fixed-parameter algorithms, and PTASs for many minimization and maximization problems, on graphs excluding a fixed minor.

1. Introduction

The deepest and likely the most important work in graph theory is the Graph Minor Theory developed by Robertson and Seymour in a series of over 20 papers spanning over 20 years. Our goal is to make this work algorithmic.

The heart of the Graph Minor Theory is a decomposition theorem [46, Theorem 1.3] capturing the structure of all graphs excluding a fixed minor. At a high level, the theorem says that every such graph can be decomposed into a collection of graphs each of which can "almost" be embedded into a bounded-genus surface, combined in a tree structure. The main result of this paper is a polynomial-time algorithm to compute such a decomposition, which we show has extensive algorithmic applications.

Most of the Graph Minor Theory consists of existential results, and some of the proofs are nonconstructive. A classic example is the celebrated proof of Wagner's Conjecture [47], which can be stated as follows: every minor-closed graph property (preserved under taking of minors) is characterized by a finite set of forbidden minors. The proof of this theorem uses the decomposition theorem mentioned above, as well as transfinite induction, and gives little insight into the finitely many forbidden minors which are proved to exist. Indeed, there is a mathematical sense in which any proof of this result must be nonconstructive [29].

Essentially the only explicitly algorithmic part of the Graph Minor Theory is a polynomial-time algorithm for testing the existence of fixed minors [44] which, combined with the proof of Wagner's Conjecture, implies the existence of a polynomial-time algorithm for deciding any minor-closed graph property. This consequence has been used to show the existence of polynomial-time algorithms for several graph problems, some of which were not previously known to be decidable [28]. However, these algorithmic results (except the minor test) are nonconstructive: we know that efficient algorithms exist, but do not know what they are. The difficulty is in determining the finite set of forbidden minors: we lack "a means of identifying the elements of the set, the cardinality of the set, or even the order of the largest graph in the set" [28].

Algorithms for H-minor-free graphs for a fixed graph H have been studied extensively; see e.g. [8, 31, 9, 35, 37]. In particular, it is generally believed that several algorithms for planar graphs can be generalized to H-minor-free graphs for any fixed H [31, 35, 37]. The decomposition theorem provides the key insight into why this might be possible: first extend an algorithm for planar graphs to handle bounded-genus graphs, then extend it to handle graphs "almost-embeddable" into bounded-genus surfaces, and finally extend it to handle tree decompositions into such graphs. However, such an approach requires an algorithm to construct the decomposition. This paper provides such an algorithm, a key stepping stone for constructing algorithms for H-minor-free graphs.

In its existential form, the graph-minor decomposition theorem has already been used to obtain many combinatorial results and the existence of many efficient algorithms, despite being published only recently. Grohe [30] proves the existence of PTASs for minimum vertex cover, min-

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

†Graduate School of Information Sciences, Tohoku University, Aramaki aza Aoba 09, Aoba-ku Sendai, Miyagi 980-8579, Japan. k_keniti@dais.is.tohoku.ac.jp

imum dominating set, and maximum independent set in H-minor-free graphs. This theorem is existential not because the algorithm requires a decomposition (though its existence is used in the analysis), but because the algorithm relies on efficient detection of minor-closed properties (which exists but is nonconstructive as mentioned above). We can modify the algorithm to rely instead on the decomposition, and therefore our work makes this result constructive. The bidimensionality theory, developed in the series [21, 19, 14, 15, 12, 16, 13, 20, 18, 17], uses the decomposition theorem to develop subexponential fixed-parameter algorithms and PTASs for a broad class of problems in H-minor-free graphs. In [12] a subexponential fixed-parameter algorithm, with running time $2^{O(\sqrt{k})}n^{O(1)}$, is developed for minimum dominating set and minimum vertex cover in H-minor-free graphs. This algorithm is conditioned on having the decomposition, and thus our decomposition algorithm removes this condition, resulting in a truly constructive result. In [18] a *parameter-treewidth bound* is established, bounding the treewidth by a small function (usually the square-root) of the optimal solution value for many problems in general H-minor-free graphs and even more problems in apex-minor-free graphs. Combined with bounded-treewidth algorithms, this bound results in many subexponential fixed-parameter algorithms, and with further ideas these results can be extended to PTASs [17]. In [16] it is shown that every minor-closed graph family with bounded local treewidth (apex-minor-free graphs) in fact has linear local treewidth, again using the decomposition theorem. This result vastly improves the running time of several PTASs based on Baker's approach (from $2^{2^{2^{O(1/\varepsilon)}}} n^{O(1)}$ to $2^{O(1/\varepsilon)}n^{O(1)}$). Other applications of the decomposition theorem include extensions of graph-minor results to countably infinite graphs [23], and the existence of a clique minor whose size is linear in the connectivity of the graph [6].

We believe that our algorithmic decomposition is a useful tool for developing efficient algorithms on H-minor-free graphs. One analogy might be to algorithms for constructing a tree decomposition in a graph of small treewidth, or even constructing a planar embedding of a planar graph. In addition to the applications listed above (using previous work), we demonstrate several algorithmic results that build upon our algorithmic decomposition theorem: polynomial-time approximation schemes for any problem satisfying a few simple conditions, approximations to treewidth, approximations to finding the largest grid minor, approximations to half-integral flow relative to fractional flow, and approximations to graph coloring. For the approximation schemes and approximate graph coloring, we develop another powerful algorithmic decomposition: every H-minor-free graph can be decomposed into any constant number k of pieces such that any $k-1$ of the pieces has bounded treewidth (where the bound depends on H and k). The proof of this decomposition result is relatively simple, showing the power of our main decomposition result. An existential version of this result was shown by DeVos et al. [22] using a complicated, and not obviously constructive, approach; here we show that a much simpler, and constructive, solution is possible using known results from an earlier paper of Grohe [30]. Even for the case $k = 2$, the result is very interesting: every H-minor-free graph is just the "sum" of two bounded-treewidth graphs. In fact this case is an algorithmic solution to a conjecture of Thomas [50]. This case also immediately leads to simple constant-factor approximation algorithm for almost every problem solvable on bounded-treewidth graphs, and by tuning k relative to $1/\varepsilon$, we often obtain a PTAS. We give general results providing PTASs for a variety of minimum and maximization problems, essentially providing a generalized Baker's approach that applies to all H-minor-free graphs, not just apex-minor-free (or planar) graphs [3, 24]. Our approximations to treewidth and grid minors exploit the minimax relation between these two quanties, leading to a combinatorial "primal-dual" type algorithm. This approach also leads us to efficient combinatorial algorithms for constructing half-integral multicommodity flows that are at most a polylogarithmic factor away from the optimal fractional multicommodity flow.

A significant approximation result in this paper is a 2-approximation algorithm for minimum graph coloring (also known as minimum chromatic number) in H-minor-free graphs. 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 ZPP = NP [26]. Even for 3-colorable graphs, the best approximation algorithm achieves a factor of $O(n^{3/14} \lg^{O(1)} n)$ [5]. 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), and the best previous approximation comes from a simple $O(|V(H)|\sqrt{\lg |V(H)|})$-approximation following from an algorithm that guarantees a coloring with $O(|V(H)|\sqrt{\lg |V(H)|})$ colors.

This problem has close connections to Hadwiger's conjecture, one of the major unsolved problems in graph theory, which can be stated as follows: every H-minor-free graph has a vertex coloring with $|V(H)|-1$ colors. Hadwiger [33] 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 [52], and therefore also true [2, 1, 38]. The case $|V(H)| = 6$ was proved by Robertson, Seymour, and Thomas [39], also using the four-color theorem. All cases $|V(H)| \geq 7$ remain unsolved. 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., [36, 51]. Thus, Hadwinger's conjecture is not resolved even up to constant factors, and the conjecture itself is only a worst-case bound. In contrast, our 2-approximation algorithm gives the best coloring, up to constant factors, for any specified H-minor-free graph (as opposed to the worst case). Furthermore, the result is algorithmic, and the approach is conceptually simple with our decomposition results in hand.

This paper is organized as follows. We start in Section 2 with a formal description of the graph-minor decomposition theorem for which we give an algorithm. Then we describe several applications of our decomposition algorithm in Section 3. Section 4 gives an overview of the main ingredients in our decomposition algorithm, while the details of our algorithm are relegated to the full paper (available on the authors' homepages).

2. Graph Minor Decomposition Theorem

This section describes the Robertson-Seymour decomposition theorem characterizing the structure of H-minor-free graphs, which we make algorithmic in this paper.

First we define the basic notion of minor. Given an edge $e = \{v, w\}$ in a graph G, the *contraction* of e in G is the result of identifying vertices v and w in G and removing all loops and duplicate edges. A graph H obtained by a sequence of such edge contractions starting from G is said to be a *contraction* of G. A graph H is a *minor* of G if H is a subgraph of some contraction of G. A graph class $\mathcal{C}$ is *minor-closed* if any minor of any graph in $\mathcal{C}$ is also a member of $\mathcal{C}$. A minor-closed graph class $\mathcal{C}$ is *H-minor-free* if $H \notin \mathcal{C}$. More generally, we use the term "H-minor-free" to refer to any minor-closed graph class that excludes some fixed graph H.

Second we define the basic notion of treewidth, introduced by Robertson and Seymour [40]. 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.

Third, we need a basic notion of embedding; see, e.g., [43, 7]. 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 [46, 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 non-planarity" 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 [52] 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 [46, 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.

As stated in [22], the proof of this theorem in [46] in fact establishes a stronger result (which also follows from our proof and algorithm):

Theorem 2.2 *The clique-sum decomposition of Theorem 2.1, written as $G_1 \oplus G_2 \oplus \cdots \oplus G_k$, has the additional property that the join set of each clique-sum between $G_1 \oplus G_2 \oplus \cdots \oplus G_{i-1}$ and G_i is a subset of the apices in G_i. Furthermore, the join set of each clique-sum involving piece G_j contains at most three vertices from the bounded-genus part of G_j.*

The main result of this paper is a polynomial-time algorithm to find the decomposition guaranteed by Theorem 2.2.

3. Algorithmic Applications of Graph-Minor Decomposition

3.1. Partition into Bounded-Treewidth Graphs

First we generalize layerwise decomposition for H-minor-free graphs, previously developed by Baker for planar graphs [3] and by Eppstein for apex-minor-free graphs [24].

Theorem 3.1 *For a 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.*

Proof: By Theorem 2.2, every H-minor-free graph can be written as a clique sum $P_1 \oplus P_2 \oplus \cdots \oplus P_\ell$ of h-almost-embeddable graphs $P_1, P_2, \ldots, P_\ell$ 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}.

First we label the vertices of each piece P_i with $k+1$ labels such that any k of the labels from the same piece induce a graph of treewidth at most $c_H k$. The *label sets* (sets of vertices with the same label) thus form a partition of the desired type for each piece P_i. Let X_i denote the apex set in piece P_i. By [30, Proposition 10], $P_i - X_i$ has linear local treewidth for fixed H, say with $f(q) = c \cdot q$. We run a bread-first search from some root vertex r_i, and assign the label to each vertex v to be the distance between r_i and v modulo $k+1$. The union of any k label sets is the disjoint union of subgraphs of $P_i - X_i$ each consisting of at most k breadth-first layers of $P_i - X_i$. By [30, Lemma 16], each of these subgraphs, and therefore the union, has treewidth at most ck. We can assign labels to the apices X_i arbitrarily (as prescribed later) and increase the treewidths by an additive constant h. Therefore the treewidth of any k label sets within P_i is at most $ck+h \leq (c+h)k$, so we set $c_H = c+h$.

Suppose by induction that $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ has a labeling with $k+1$ labels such that any k label sets induce a graph of treewidth at most $c_H k$. We have already proved the base case of $i = 1$. 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 [19, Lemma 3], for any two graphs G' and G'', $\text{tw}(G' \oplus G'') \leq \max\{\text{tw}(G'), \text{tw}(G'')\}$. Thus, the treewidth of any j label sets in $(P_1 \oplus P_2 \oplus \cdots \oplus P_i) \oplus P_{i+1}$ is at most the maximum of the treewidth of the j label sets within $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ and the treewidth of the j label sets within P_{i+1}. The latter is at most $c_H k$ as argued above, and the former is at most $c_H k$ by the induction hypothesis. Therefore the label sets form the desired partition.

We can obtain an edge partition in parallel to a vertex partition by a similar inductive construction. First we assign the label of each edge in $P_i - X_i$ to be the label of the endpoint closest to the root r_i, in the vertex labeling of $P_i - X_i$. To each remaining edge in P_i, with one or both endpoints in X_i, we assign the label of an endpoint in X_i (choosing the endpoint arbitrarily if there is a choice). As before, the treewidth of any k label sets within P_i is at most $c_H k$. Then we combine these labelings as follows. Suppose by induction that $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ has a vertex and an edge labeling with $k+1$ labels such that any k vertex label sets or any k edge label sets induce a graph of treewidth at most $c_H k$. We have already proved the base case of $i = 1$. We merge the vertex labelings of $P_1 \oplus P_2 \oplus \cdots \oplus P_i$ and P_{i+1} as before (and thus we obtain the same vertex labeling as before). We merge the edge labelings as follows: whenever an edge in P_{i+1} has exactly one endpoint in the join set J_{i+1}, we use the the inductive label assigned to that endpoint by $P_1 \oplus P_2 \oplus P_i$; and whenever an edge has both endpoints in the join set J_{i+1}, we use the inductive label assigned to the edge by $P_1 \oplus P_2 \oplus P_i$. This labeling is a particular decision for the arbitrary choices made by edges with both endpoints in $X_{i+1} \supseteq J_{i+1}$. Thus, every edge connecting J_{i+1} to $P_{i+1} - J_{i+1}$ is assigned the label of the endpoint in J_{i+1}. Let L be the set of all vertices that have one of the desired k labels. We claim that the subgraph of $(P_1 \oplus P_2 \oplus \cdots \oplus P_i) \oplus P_{i+1}$ induced by any k edge label sets is itself a clique-sum of two graphs with join set $J_{i+1} \cap L$. (In this subgraph, any vertex in J_{i+1} assigned the excluded label has no incident edges connecting to $P_{i+1} - J_{i+1}$, so we can remove this vertex from P_{i+1} and J_{i+1} in the clique-sum, and the remaining vertices in the join set form a clique.) The two summed graphs have treewidth at most $c_H k$ by induction, and thus the clique-sum has treewidth at most $c_H k$. Therefore the label sets form the desired partition.

The construction of the label sets runs in linear time given the decomposition from Theorem 2.2, for a polynomial overall time bound. □

Approximate coloring. Applying Theorem 3.1 for $k = 2$, and because minimum graph coloring can be solved optimally in graphs of bounded treewidth, we obtain the following important theorem:

Theorem 3.2 *In H-minor-free graphs, there is a polynomial-time 2-approximation for minimum graph coloring.*

This approximation factor is near optimal because minimum graph coloring is hard to approximate better than $4/3$ even in planar graphs. The factor improves over a trivial $O(|V(H)|\sqrt{\lg |V(H)|})$-approximation arising from a $O(|V(H)|\sqrt{\lg |V(H)|})$-coloring that follows from average degree bounds in H-minor-free graphs [36, 51]. Our technique can be generalized to obtain approximation algorithms for many graph problems, as developed in the following sections.

The same technique as coloring can be applied to many other problems. An example of a maximization problem is the notorious *dense k-subgraph* problem, for which the best approximation known is $O(n^{1/3-\varepsilon})$ [27]. We obtain a 2-approximation for H-minor-free graphs.

3.2. Approximation Algorithms for Minimization Problems

We start with a simple but very general constant-factor approximation, which in some cases (such as minimum graph coloring) is near optimal:

Theorem 3.3 *Suppose a minimization problem P on graphs has the following properties:*

1. *there is a polynomial-time algorithm solving P on graphs of bounded treewidth;*
2. *the value of the optimal solution for P never increases when removing vertices (respectively, edges); and*
3. *given a partition of the vertices (respectively, edges) of a graph G into two sets, a solution to each of the induced subgraphs of G can be merged in polynomial time into a solution for G of value at most α times the sum of the two solution values.*

For any fixed H, there is a polynomial-time (2α)-approximation algorithm for problem P in H-minor-free graphs.

See the full paper for this and other omitted proofs.

From Theorem 3.3 we obtain easy $O(1)$-approximations for many graph problems. For some problems (such as graph coloring) this is the best result known, while for some such problems PTASs are possible. For example, Theorem 3.3 gives a 4-approximation for minimum color sum, but below we obtain a PTAS.

Next we develop a PTAS for many of these minimization problems.

Theorem 3.4 *Suppose a minimization problem P satisfies the following properties:*

1. *there is a polynomial-time algorithm solving P on graphs of bounded treewidth;*
2. *given a partition of the vertices (respectively, edges) of a graph G into two sets S_1 and S_2,*
   (a) *there are solutions F_1 and F_2 to P on the induced subgraphs $G[S_i]$ (e.g., $F_i = \mathrm{OPT}(G) \cap S_i$) such that the total value of the two solutions is at most the optimal solution value for G; and*
   (b) *given solutions for $G[S_1]$ and $G[S_2]$ can be merged in polynomial time into a solution for G of value at most $1+\alpha$ times the first solution value plus $1/\alpha$ times the second solution value, for any $0 < \alpha \leq 1$. (This condition is satisfied in particular if the merged solution value is at most the sum of the two solution values.)*

For any fixed H and any $0 < \varepsilon \leq 1$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm for problem P in H-minor-free graphs.

Proof: We apply Theorem 3.1 with $k + 1 = 4/\varepsilon^2$ to obtain a partition of a given graph G into sets $S_1, S_2, \ldots, S_{k+1}$ of vertices or edges. Let G_i be the subgraph of G induced by $S_1 \cup S_2 \cup \cdots \cup S_{i-1} \cup S_{i+1} \cup \cdots \cup S_{k+1}$. Each G_i has bounded treewidth and thus we can compute the optimal solution $\mathrm{OPT}(G_i)$ in polynomial time. Similarly, $G[S_i]$ has bounded treewidth, and we can compute its optimal solution

$\mathrm{OPT}(G[S_i])$ in polynomial time. The approximation algorithm merges $\mathrm{OPT}(G_i)$ and $\mathrm{OPT}(G[S_i])$ with $\alpha = \varepsilon/2$, for each i, and returns the best such solution.

By repeated application of Property 2(a), there is a solution F_i to P on each induced subgraph $G[S_i]$ such that the total value of these solutions is at most $\mathrm{OPT}(G)$. Thus, for some i, F_i has weight at most $\mathrm{OPT}(G)/(k+1)$. Therefore, $\mathrm{OPT}(G[S_i]) \leq \mathrm{OPT}(G)/(k+1)$. Also, by Property 2(a), $\mathrm{OPT}(G_i) \leq \mathrm{OPT}(G)$. The value of the constructed merged solution for this value of i is $(1+\alpha)\,\mathrm{OPT}(G_i) + (1/\alpha)\,\mathrm{OPT}(G[S_i])$ which is at most $(1+\alpha)\,\mathrm{OPT}(G) + (1/\alpha)\,\mathrm{OPT}(G)/(k+1) = (1+\alpha+(1/\alpha)/(k+1))\,\mathrm{OPT}(G) = (1+\varepsilon)\,\mathrm{OPT}(G)$ by our choices of α and $k+1$. $\square$

This result is very general and applies to a wide variety of problems to which Baker's approach applies, such as vertex cover. One particularly interesting application is the well-studied variation of graph coloring called *minimum color sum* [4, 26, 34], where the goal is to find a (vertex or edge) coloring with positive integers with minimum total value. We can use the bounded-treewidth algorithm of Halldórsson and Kortsarz [34], and the merging strategy of introducing each color from the second solution after each group of $1/\alpha$ colors from the first solution.

Corollary 3.5 *There is a PTAS for minimum color sum (of vertices or edges) on H-minor-free graphs.*

A *linear kernelization* of a minimization problem on weighted graphs is a polynomial-time algorithm that, given a weighted graph G, constructs a weighted graph G' such that $\mathrm{OPT}(G') \leq \mathrm{OPT}(G)$, $\mathrm{OPT}(G')$ is at least β times the total weight of the vertices (respectively, edges) of G', and any solution for G' can be converted in polynomial time to a solution for G with value larger by at most $\mathrm{OPT}(G) - \mathrm{OPT}(G')$.

Theorem 3.6 *Suppose that a minimization problem P has a linear kernelization, can be solved in polynomial time on graphs of bounded treewidth, and the value of the optimal solution for P never increases when removing vertices (respectively, edges). Also suppose that, given a partition of the vertices (respectively, edges) of a graph G into two sets S_1 and S_2, and given a solution for $G[S_1]$, we can compute in polynomial time a solution for G of value at most $1+\alpha$ times the solution value for $G[S_1]$ plus $1/\alpha$ times the total weight of S_2, for any $0 < \alpha \leq 1$. (This condition is satisfied in particular if the solution value for G is at most the solution value for $G[S_1]$ plus $O(1)$ times the total weight of $G[S_2]$.) For any fixed H and any $0 < \varepsilon \leq 1$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm for problem P in H-minor-free graphs.*

This theorem can also be applied to minimum color sum, without kernelization because $\mathrm{OPT}(G) \geq |V(G)|$, using the constant average degree bound (for fixed H) of [36, 51] to color $G[S_2]$ with $O(1)$ colors, and using the same bounded-treewidth algorithm and merging technique as above.

3.3. Approximation Algorithms for Maximization Problems

In this section we develop PTASs for a broad class of maximization problems. A graph property π is *hereditary* if every induced subgraph of a graph with the property also has the property. The *maximum (weighted) induced subgraph problem* for a graph property π, MISP(π), is to find the largest (maximum-weight) set of vertices in a graph G that induce a subgraph with property π; similarly, the EMISP(π) problem is to find the largest (maximum-weight) set of edges that induce a subgraph with property π (and the hereditary property of π is in terms of edges instead of vertices). Examples of MISP-type problems include finding the maximum induced subgraph that is chordal, acyclic, without cycles of a specified length, without edges (independent set), of maximum degree $r \geq 1$, bipartite, a clique, or planar [53]. Yannakakis [53] has shown that these examples of MISP are all NP-complete, and for all except the last example, NP-complete even when restricted to planar graphs [53]. Another interesting example is maximum cut, which is equivalent to EMISP(π) where π is the property of the graph being bipartite; our PTAS for this problem is an interesting complement to the polynomial-time algorithm for maximum cut in planar graphs [32].

Theorem 3.7 *For any hereditary graph property π that can be solved in polynomial time on graphs of bounded treewidth, for any graph H, and for any $\varepsilon > 0$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm for MISP(π) and EMISP(π) on H-minor-free graphs.*

This result generalizes results of Chen [11] for $K_{3,3}$-minor-free and K_5-minor-free graphs, Demaine et al. [19] for single-crossing-minor-free graphs, and Grohe [30] for independent set.

Our approach can also obtain a PTAS for *maximum P-matching* [3], where the goal is to find the maximum number of vertex-disjoint induced subgraphs isomorphic to a fixed graph P. This problem includes the special cases of maximum triangle matching and maximum tile salvage.

3.4. Subexponential Fixed-Parameter Algorithms

The newly developing theory of bidimensional graph problems, developed in a series of papers [21, 19, 14, 15, 12, 16, 13, 20, 18, 17], provides general techniques for designing efficient fixed-parameter algorithms and approximation algorithms for NP-hard graph problems in broad classes of graphs. This theory applies to graph problems that are *bidimensional* in the sense that (1) the solution value for $k \times k$ "grid-like" graphs grows with k, typically as $\Omega(k^2)$, and (2) the solution value only goes down when contracting edges and optionally when deleting edges. Examples of such problems include feedback vertex set, vertex cover, minimum maximal matching, face cover, a series of vertex-removal parameters, dominating set, edge dominating set, R-dominating set, connected dominating set, connected edge dominating set, connected R-dominating set, and unweighted TSP tour (a walk visiting all vertices).

Bidimensional problems are divided into *contraction-bidimensional* and *minor-bidimensional* problems according to whether the solution value only goes down when deleting edges. The bidimensionality theory obtains subexponential fixed-parameter algorithms, with typical running time $2^{O(\sqrt{k})}n^{O(1)}$, for minor-bidimensional problems in all H-minor-free graphs [12] and for contraction-bidimensional problems in all apex-minor-free graphs. However, the only subexponential fixed-parameter algorithm for a contraction-bidimensional problem on general H-minor graphs is a complicated algorithm for dominating set and several variants [12], and this algorithm assumes that the Robertson-Seymour decomposition is given.

Our algorithm for Theorem 2.2 fills this gap, providing a fundamental building block for future development of subexponential fixed-parameter algorithms on H-minor-free graphs. In particular, we obtain the following result from [12, Theorem 4.4]:

Theorem 3.8 *For any fixed H, there is an algorithm that finds a dominating set of size at most k in a given H-minor-free graph in $2^{O(\sqrt{k})}n^{O(1)}$ time.*

3.5. Approximating Treewidth

The algorithm of this section and the next essentially form a kind of combinatorial primal-dual algorithm, in which we effectively use the following minimax relation between minimum treewidth and maximum grid minors:

Theorem 3.9 [18] *For any fixed graph H, every H-minor-free graph of treewidth w has an $\Omega(w) \times \Omega(w)$ grid as a minor.*

Theorem 3.10 *For any fixed H, there is a polynomial-time $O(1)$-approximation algorithm for computing a tree decomposition of minimum width in an H-minor-free graph.*

Proof: The algorithm proceeds as follows. First we compute the decomposition of G from Theorem 2.2 into a clique-sum $P_1 \oplus P_2 \oplus \cdots P_k$ of h-almost-embeddable graphs. For each P_i, we form a bounded-genus graph B_i by starting from the bounded-genus part of P_i (excluding all apices and vortices), and forming a cycle on the vertices that attached to each vortex. This graph is referred to as $\tilde{G}$ in [18], and in [18, Lemmas 4.1–4.5] it is shown that $\mathrm{tw}(B_i) = O(\mathrm{tw}(G))$ for all i and $\mathrm{tw}(B_i) = \Omega(\mathrm{tw}(G))$ for some i. Therefore for approximating treewidth it suffices to compute the treewidth of each B_i, and then take the maximum. Of course, this process is complicated by requiring an actual tree decomposition, not just the value of treewidth.

To compute a tree decomposition of the bounded-genus graph B_i of width $O(\mathrm{tw}(B_i))$, we find a minimum-length noncontractible noose using the algorithm of [7]. By [12, Lemma A.1], if a bounded-genus graph has face-width w, then it has an $\Omega(w) \times \Omega(w)$ grid minor, and thus in particular it has treewidth $\Omega(w)$. Thus the (minimum) length of the noncontractible noose, which is the face-width of B_i, is $O(\mathrm{tw}(B_i))$. We remove all vertices from the noose, reducing the genus of the graph by 1, and possibly disconnecting the graph. We call each connected component a *reduced graph*. Now we repeat this process by, in each round, finding and removing a minimum-length noncontractible noose in each reduced graph that is not already planar. The process stops when all reduced graphs are planar. Because we only remove vertices, the treewidth of each reduced graph is at most $\mathrm{tw}(B_i)$. Now we can apply a 1.5-approximation for minimum-width tree decomposition on planar graphs [49] to obtain a tree decomposition of each reduced graph with width $O(\mathrm{tw}(B_i))$.

Now we reverse the noose-removal process, in each round adding the removed nooses from each of the reduced graphs, and recombining the reduced graphs in the reverse order as before. We combine the tree decompositions by placing all vertices of the added nooses into all bags of the incident reduced graphs. Because each reduced graph is incident to at most one noose during a round, and each noose has length $O(\mathrm{tw}(B_i))$, this addition increases the width of the tree decomposition by $O(\mathrm{tw}(B_i))$ during each round. The number of rounds is at most the genus g of B_i, because the genus of each reduced graph reduced by at least 1 during each round of noose removal. Therefore we obtain a tree decomposition of B_i of width $O((g+1)\,\mathrm{tw}(B_i))$. Because $g \le h$, this width is $O(h\,\mathrm{tw}(B_i)) = O(\mathrm{tw}(B_i))$.

Next we show how to add the vortices back to B_i, forming a new graph B_i', while preserving the tree decomposition. Let $U_i = \{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$ be the cyclically ordered vertices of B_i at which a vortex was attached but replaced by a cycle. For each u_i^j that occurs in bag $\mathcal{B}$ in the tree decomposition of B_i, we add to $\mathcal{B}$ the corresponding bag $\mathcal{B}_{u_i^j}$ from the path decomposition of vortex G_i''. The resulting bags form a tree decomposition of B_i' because $\{u_i^1, u_i^2, \ldots, u_i^{m_i}\}$ are connected in a path in B_i. By charging the $\le h+1$ added vertices to the occurrence of u_i^j that triggered the addition, each bag increases in size by a factor at most $h+1$ for each of the h vortices. Thus the width of this tree decomposition of B_i' is $O(h^2\,\mathrm{tw}(B_i)) = O(\mathrm{tw}(B_i))$,

It is easy to add the apices back to B_i': simply place each apex in every bag. The width of the tree decomposition increases by at most h. Now we have a tree decomposition of the original pieces P_i of width $O(\mathrm{tw}(B_i)) = O(\mathrm{tw}(P_i))$. We combine these tree decompositions according to the clique-sum decomposition. For each clique-sum between two graphs, we find a bag in each decomposition containing all nodes of the join set (such a bag must exist because the join set forms a clique), and then we connect the tree nodes of these two bags by adding an edge, connecting the two tree decompositions. The width of the resulting decomposition is the maximum of the widths of the two given tree decompositions. Therefore we obtain a tree decomposition of width $O(\max_i \mathrm{tw}(P_i)) = O(\mathrm{tw}(G))$. □

Recently, a noncombinatorial approach based on linear programming has been developed to obtain an $O(1)$-approximation to treewidth in H-minor-free graphs [25].

3.6. Approximating Grid Minors

The primal-dual nature of the algorithm from the previous section allows us to construct grid minors as follows. The following result improves the fixed-parameter algorithm of Robertson and Seymour [44] for constructing $k \times k$ grid minors for fixed k, to the case of arbitrary k in H-minor-free graphs.

Theorem 3.11 *For any fixed H, there is a polynomial-time $O(1)$-approximation algorithm for computing the maximum $k \times k$ grid minor in an H-minor-free graph.*

Our construction uses an important concept from the existential result relating treewidth and grid minors [18]. Consider a graph G decomposed into a clique-sum $P_1 \oplus P_2 \oplus \cdots P_k$ of almost-embeddable graphs. The *approximation graph* A_i of the almost-embeddable graph P_i is formed from the bounded-genus part of P_i (thus excluding all apices and vortices) by (a) removing all vertices in the bounded-genus part that were attached to vortices, (b) removing all virtual edges (edges in P_i that are not in G), and (c) replacing some of these edges as follows. For each clique sum involving P_i with the property that the join set W contains at least two vertices not already removed (note that there can be at most three such vertices), we do the following: (i) if W contains exactly two vertices not already removed, we add an edge between these two vertices; (ii) if W contains three vertices not already removed and there is more than one clique sum whose join set contains these three vertices, we add a triangle of edges between these three vertices; (iii) if W contains three vertices not already removed and there is only one clique sum whose join set contains these three vertices, we add a new vertex v inside the virtual triangle they form on the surface and then add an edge connecting v to each of the three vertices.

Proof: The algorithm proceeds as follows. First we compute the decomposition of G according to Theorem 2.2. Second we construct the approximation graph A_i of each summand in the clique-sum decomposition. By [18, Lemmas 4.1–4.5], each A_i is a minor of G and $\max_i \mathrm{tw}(A_i) = \Theta(\mathrm{tw}(G))$. Thus it suffices to find an approximately largest $k \times k$ grid minor in each A_i, and return the largest such minor, to find an approximately largest $k \times k$ grid minor in G.

For each A_i, we compute the shortest noncontractible noose using the algorithm of [7], whose length is the face-width of A_i. If the face-width of A_i is at least $\mathrm{tw}(A_i)/(g+1)$, where g is the genus of A_i, then by [12, Lemma A.1], we can construct an $\Omega(\mathrm{tw}(A_i/(g+1)) \times \Omega(\mathrm{tw}(A_i/(g+1))$ grid minor. Otherwise, we remove the vertices of the noncontractible noose, decreasing the treewidth of A_i by at most $\mathrm{tw}(A_i)/(g+1)$. The resulting graph A_i' may be disconnected, but $\mathrm{tw}(A_i')$ is the maximum treewidth among the connected components. We recurse on each connected component, and return the largest grid minor found. At the base of the recursion, we have a planar graph: we compute a branch decomposition of minimum width w [49] and from that we compute an $\Omega(w) \times \Omega(w)$ grid minor using the algorithm implicit in the proof of [48, Theorem 6.2]. The recursion has depth at most g because each noose removal decreases the genus by at least 1. Thus some graph in a leaf of the recursion has treewidth at least $\mathrm{tw}(A_i) - g\,\mathrm{tw}(A_i)/(g+1) = \mathrm{tw}(A_i)/(g+1)$, so it has branchwidth $\Omega(\mathrm{tw}(A_i)/(g+1)) = \Omega(\mathrm{tw}(A_i))$. □

3.7. Half-Integral Versus Fractional Multicommodity Flow

Chekuri, Khanna, and Shephard [10] proved that, for planar graphs, the gap between the optimal half-integral multicommodity flow and the optimal fractional multicommodity flow is at most a polylogarithmic factor. Furthermore, they gave a combinatorial algorithm for constructing a polylogarithmic-approximate multicommodity flow in planar graphs. Both the bound on the gap and the combinatorial algorithm are based on the existence of an $\Omega(w) \times \Omega(w)$ grid minor in a planar graph of treewidth w; the algorithm essentially constructs such a grid minor.

As mentioned by Chekuri et al. [10], the existence of an $\Omega(w) \times \Omega(w)$ grid minor in an H-minor-free graph of treewidth w (Theorem 3.9) can be used to generalize the bound on the gap between half-integral and fractional multicommodity flows. However, so far, this result is just existential. Using our constant-factor approximation algorithm from Theorem 3.11 for finding the largest grid minor, we can extend the combinatorial algorithm as well:

Theorem 3.12 *For any fixed H, there is a polynomial-time algorithm computing a half-integral multicommodity flow in a given H-minor-free graph that is within a polylogarithmic factor of the optimal fractional multicommodity flow.*

Chekuri et al. [10] also gave a combinatorial proof of the result that, for planar graphs, the gap between the maximum flow and the minimum cut in product multicommodity flow (and thus uniform multicommodity flow) instances is at most a constant factor. The latter result was proved before by Klein, Plotkin, and Rao for H-minor-free graphs using primal-dual methods [35], and has many applications in embeddings of H-minor-free graphs. Again our algorithm from Theorem 3.11 for finding grids in H-minor-free graphs enables us to make this existential result algorithmic and construct such a multicommodity minimum cut in polynomial time.

4. Overview of Our Decomposition Algorithm

At a high level, our algorithm follows the proof of Theorem 2.1, in [46, Theorem 1.3]. However, we skip several of the existential steps in favor of a more algorithmic approach.

The proof of Theorem 2.1 splits into two main components: handling each term in the clique-sum, and putting these terms together. Both components are difficult. The first component uses another deep structural theorem about H-minor-free graphs. This theorem is described in [46, Theorem 3.1], and it is used, e.g., in the proof of Wagner's Conjecture [47]. Roughly, if we eliminate the "tree structure" of clique sums in Theorem 2.1, and concentrate on the internal structure of one of the "nodes" of the tree, the local structure of G has a large "wall" (a k-*wall* is a particular subdivision of a $k \times k$ grid) and, up to 3-cuts, it is h-almost-embeddable into some surface in which H cannot be embedded, except that we cannot bound the pathwidth of vortices. The second component uses a nontrivial tree-decomposition theorem [42, Theorem 11.1].

We sketch how to obtain the structure in the first component. If the treewidth of the graph is small, the problem is relatively easy, so we focus on the case of large treewidth. The main result of [48] says that there exists a function $f(k)$ such that, if the treewidth of a graph G is at least $f(k)$, then G has a $k \times k$ grid minor. For our algorithmic purposes, we shall use a k-wall instead of a $k \times k$ grid-minor. By a theorem of [44], we can detect a k-wall if one exists.

We guess the correct set X of at most h apex vertices (by trying all possible choices). Then $G - X$ is apex-free h-almost-embeddable in some surface in which H cannot be embedded, except that we cannot bound the width of vortices. Also we can find a large wall H_0.

Similar to the existential proof of Robertson and Seymour, our goal is to make a sequence of pairs (H_i, Σ_i) such that, at each stage, we increase the genus at the cost of sacrificing a small amount of the wall H_i, while keeping the large face-width. Our initial pair is (H_0, Σ_0) where Σ_0 is the sphere. Every graph with high face-width that can be drawn in a surface in which H can also be drawn has an H-minor, by the result of [41]. Hence the genus-increasing process stops in $O(|H|^2)$ steps or so.

How does the rest of the graph attach to H_i? It turns out that, except for a bounded number of disks which become vortices, the rest of the graph is just attached to disks of H_i and drawn in these disks up to 3-cuts. Hence the remaining challenge is to find the structure of how the rest of the graph attaches to these disks of H_i. There are three ingredients that we need to detect: (1) How to find a handle? (2) How to find a crosscap? (3) How to find a vortex? By the deep thereoms [46, Theorem 8.1] and [45, Theorem 1.1], it turns out that, if there is a handle, then there must be a large "jump" in (H_i, Σ_i). Also, if there is no jump, then [45, Theorem 1.1] says that crosscaps and vortices are contained in small (bounded-diameter) disks of (H_i, Σ_i). We use these properties to efficiently detect handles, crosscaps, and vortices.

In the second component, we suppose that we can already detect G as h-almost-embeddable in some surface in which H cannot be embedded, except that we cannot bound the pathwidth of vortices. Now our goal is to translate this into the structure of Theorem 2.1 [46, Theorem 1.3]. This translation can be done by using the highly nontrivial theorem [42, Theorem 11.3]. Roughly, suppose we are given the structure of [46, Theorem 3.1]. If some bag in one vortex has large treewidth, then we repeatedly apply the algorithm of [46, Theorem 3.1] to this bag until there is no large bag in the resulting structure. [42, Theorem 11.3] gurantees that we can control the location of a bounded number of vertices in the resulting structure of this bag. This means that we can obtain the structure of [46, Theorem 3.1] in such a way that the specified vertex set is always contained in the set of apex vertices. This property allows us to extend the tree decomposition, and the resulting tree decompostion is the desired structure.

Acknowledgments. We thank Neil Robertson, Paul Seymour, and Robin Thomas for many helpful discussions about graph minors.

References

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

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

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

[4] A. Bar-Noy, M. Bellare, M. M. Halldórsson, H. Shachnai, and T. Tamir, *On chromatic sums and distributed resource allocation*, Information and Computation, 140 (1998), pp. 183–202.

[5] A. Blum and D. Karger, *An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs*, Information Processing Letters, 61 (1997), pp. 49–53.

[6] T. Böhme, K. Kawarabayashi, J. Maharry, and B. Mohar, *Linear connectivity forces large complete bipartite minors*. Manuscript, October 2004. http://www.dais.is.tohoku.ac.jp/~k_keniti/KakNew8.ps.

[7] S. Cabello and B. Mohar, *Finding shortest non-contractible and nonzero-homologous cycles for topologically embedded graphs*. Manuscript, 2004.

[8] M. Charikar and A. Sahai, *Dimension reduction in the l_1 norm*, in Proceedings of the 43th Annual Symposium on Foundations of Computer Science (FOCS'02), 2002, pp. 551–560.

[9] C. Chekuri, A. Gupta, I. Newman, Y. Rabinovich, and S. Alistair, *Embedding k-outerplanar graphs into ℓ_1*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03), 2003, pp. 527–536.

[10] C. Chekuri, S. Khanna, and F. B. Shepherd, *Edge-disjoint paths in planar graphs*, in Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS 2004), 2004, pp. 71–80.

[11] Z.-Z. Chen, *Efficient approximation schemes for maximization problems on $K_{3,3}$-free or K_5-free graphs*, Journal of Algorithms, 26 (1998), pp. 166–187.

[12] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, Journal of the ACM. To appear. A preliminary version appears in *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms*, January 2004, pages 823–832.

[13] ———, *Bidimensional parameters and local treewidth*, SIAM Journal on Discrete Mathematics, 18 (2004), pp. 501–511.

[14] ———, *Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs*, ACM Transactions on Algorithms, 1 (2005).

[15] E. D. Demaine and M. Hajiaghayi, *Diameter and treewidth in minor-closed graph families, revisited*, Algorithmica, 40 (2004), pp. 211–215.

[16] ———, *Equivalence of local treewidth and linear local treewidth and its algorithmic applications*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA'04), January 2004, pp. 833–842.

[17] E. D. Demaine and M. Hajiaghayi, *Bidimensionality: New connections between FPT algorithms and PTASs*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, January 2005, pp. 590–601.

[18] ———, *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), Vancouver, January 2005, pp. 682–689.

[19] E. D. Demaine, M. Hajiaghayi, N. Nishimura, P. Ragde, and D. M. Thilikos, *Approximation algorithms for classes of graphs excluding single-crossing graphs as minors*, Journal of Computer and System Sciences, 69 (2004), pp. 166–195.

[20] E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos, *The bidimensional theory of bounded-genus graphs*, SIAM Journal on Discrete Mathematics. To appear.

[21] E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos, *Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors*, Algorithmica, 41 (2005), pp. 245–267.

[22] M. DeVos, G. Ding, B. Oporowski, D. P. Sanders, B. Reed, P. Seymour, and D. Vertigan, *Excluding any graph as a minor allows a low tree-width 2-coloring*, Journal of Combinatorial Theory, Series B, 91 (2004), pp. 25–41.

[23] R. Diestel and R. Thomas, *Excluding a countable clique*, Journal of Combinatorial Theory, Series B, 76 (1999), pp. 41–67.

[24] D. Eppstein, *Diameter and treewidth in minor-closed graph families*, Algorithmica, 27 (2000), pp. 275–291.

[25] U. Feige, M. Hajiaghayi, and J. R. Lee, *Improved approximation algorithms for minimum-weight vertex separators*, in Proceedings of the 37th ACM Symposium on Theory of Computing (STOC 2005), Baltimore, May 2005, pp. 563–572.

[26] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, Journal of Computer and System Sciences, 57 (1998), pp. 187–199.

[27] U. Feige, G. Kortsarz, and D. Peleg, *The dense k-subgraph problem*, Algorithmica, 29 (2001), pp. 410–421.

[28] M. R. Fellows and M. A. Langston, *Nonconstructive tools for proving polynomial-time decidability*, Journal of the ACM, 35 (1988), pp. 727–739.

[29] H. Friedman, N. Robertson, and P. Seymour, *The metamathematics of the graph minor theorem*, in Logic and combinatorics (Arcata, Calif., 1985), vol. 65 of Contemp. Math., Amer. Math. Soc., Providence, RI, 1987, pp. 229–261.

[30] M. Grohe, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), pp. 613–632.

[31] A. Gupta, I. Newman, Y. Rabinovich, and S. Alistair, *Cuts, trees and ℓ_1-embeddings of graphs*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS'99), 1999, pp. 399–409.

[32] F. Hadlock, *Finding a maximum cut of a planar graph in polynomial time*, SIAM Journal on Computing, 4 (1975), pp. 221–225.

[33] H. Hadwiger, *Über eine Klassifikation der Streckenkomplexe*, Vierteljschr. Naturforsch. Ges. Zürich, 88 (1943), pp. 133–142.

[34] M. M. Halldórsson and G. Kortsarz, *Tools for multicoloring with applications to planar graphs and partial k-trees*, Journal of Algorithms, 42 (2002), pp. 334–366.

[35] P. N. Klein, S. A. Plotkin, and S. Rao, *Excluded minors, network decomposition, and multicommodity flow*, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC'93), 1993, pp. 682–690.

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

[37] S. A. Plotkin, S. Rao, and W. D. Smith, *Shallow excluded minors and improved graph decompositions*, in Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'94), 1994, pp. 462–470.

[38] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, *The four-colour theorem*, Journal of Combinatorial Theory, Series B, 70 (1997), pp. 2–44.

[39] N. Robertson, P. Seymour, and R. Thomas, *Hadwiger's conjecture for K_6-free graphs*, Combinatorica, 13 (1993), pp. 279–361.

[40] N. Robertson and P. D. Seymour, *Graph minors. II. Algorithmic aspects of tree-width*, Journal of Algorithms, 7 (1986), pp. 309–322.

[41] N. Robertson and P. D. Seymour, *Graph minors. VII. Disjoint paths on a surface*, Journal of Combinatorial Theory, Series B, 45 (1988), pp. 212–254.

[42] N. Robertson and P. D. Seymour, *Graph minors. X. Obstructions to tree-decomposition*, Journal of Combinatorial Theory Series B, 52 (1991), pp. 153–190.

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

[44] ———, *Graph minors. XIII. The disjoint paths problem*, Journal of Combinatorial Theory, Series B, 63 (1995), pp. 65–110.

[45] ———, *Graph minors. XV. Giant steps*, Journal of Combinatorial Theory, Series B, 68 (1996), pp. 112–148.

[46] ———, *Graph minors. XVI. Excluding a non-planar graph*, Journal of Combinatorial Theory, Series B, 89 (2003), pp. 43–76.

[47] ———, *Graph minors. XX. Wagner's conjecture*, Journal of Combinatorial Theory, Series B, 92 (2004), pp. 325–357.

[48] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, Journal of Combinatorial Theory, Series B, 62 (1994), pp. 323–348.

[49] P. D. Seymour and R. Thomas, *Call routing and the ratcatcher*, Combinatorica, 14 (1994), pp. 217–241.

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

[51] A. Thomason, *The extremal function for complete minors*, Journal of Combinatorial Theory, Series B, 81 (2001), pp. 318–338.

[52] K. Wagner, *Über eine Eigenschaft der eben Komplexe*, Deutsche Math., 2 (1937), pp. 280–285.

[53] M. Yannakakis, *Node- and edge-deletion NP-complete problems*, in Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, CA, 1978), ACM press, New York, 1978, pp. 253–264.