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 $|Z'| \le \Theta$ 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, $|(B \cap Z) \cup (A \cap B)| \le \Theta$. 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 $|(Z \cap A) \cup (A \cap B)| \le \Theta$ and $|(Z \cap B) \cup (A \cap B)| \le \Theta$. 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.