Contraction Decomposition in H-Minor-Free Graphs and Algorithmic Applications

Erik D. Demaine
MIT CSAIL
32 Vassar St.
Cambridge, MA 02139, USA
edemaine@mit.edu

MohammadTaghi Hajiaghayi[*]
A.V. Williams Building
University of Maryland
College Park, MD 20742, USA
hajiagha@cs.umd.edu

Ken-ichi Kawarabayashi[†]
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
k_keniti@nii.ac.jp

ABSTRACT

We prove that any graph excluding a fixed minor can have its edges partitioned into a desired number k of color classes such that contracting the edges in any one color class results in a graph of treewidth linear in k. This result is a natural finale to research in contraction decomposition, generalizing previous such decompositions for planar and bounded-genus graphs, and solving the main open problem in this area (posed at SODA 2007). Our decomposition can be computed in polynomial time, resulting in a general framework for approximation algorithms, particularly PTASs (with $k \approx 1/\varepsilon$), and fixed-parameter algorithms, for problems closed under contractions in graphs excluding a fixed minor. For example, our approximation framework gives the first PTAS for TSP in weighted H-minor-free graphs, solving a decade-old open problem of Grohe; and gives another fixed-parameter algorithm for k-cut in H-minor-free graphs, which was an open problem of Downey et al. even for planar graphs.

To obtain our contraction decompositions, we develop new graph structure theory to realize virtual edges in the clique-sum decomposition by actual paths in the graph, enabling the use of the powerful Robertson–Seymour Graph Minor decomposition theorem in the context of edge contractions (without edge deletions). This requires careful construction of paths to avoid blowup in the number of required paths beyond 3. Along the way, we strengthen and simplify contraction decompositions for bounded-genus graphs, so that the partition is determined by a simple radial ball growth independent of handles, starting from a set of vertices instead of just one, as long as this set is tight in a certain sense. We show that this tightness property holds for a constant number of approximately shortest paths in the surface, introducing several new concepts such as dives and rainbows.

[*]Research supported in part by NSF CAREER award CCF-1053605.

[†]Research supported in part by Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research, by C & C Foundation, by Kayamori Foundation, and by Inoue Research Award for Young Scientists.

STOC'11, June 6–8, 2011, San Jose, California, USA.

Categories and Subject Descriptors

F.2.0 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*General*

General Terms

Algorithms, Theory

Keywords

Decomposition, Graph Contraction, Graph Minor

1. INTRODUCTION

Graph decompositions—partitioning of graphs into smaller pieces—is a fundamental way to design graph algorithms. One of the most famous such decompositions is Lipton and Tarjan's divide-and-conquer separator decomposition for planar graphs [30], generalized to arbitrary graphs via sparsest cut [3, 29]. The main technique in these decompositions is to find relatively small cuts in the graph that minimize the interaction between the pieces. To make the pieces relatively small, the decompositions cut the graph into many pieces.

An alternative approach is to partition the graph into a small number of computationally simpler (but not necessarily small) pieces, allowing large interaction between the pieces. For instance, we can solve many optimization problems efficiently on graphs of bounded treewidth. If a graph can be partitioned into a small number s of bounded-treewidth pieces, then in many cases, each piece gives a lower/upper bound on the optimal solution for the entire graph, so solving the problem exactly in each piece gives an s-approximation to the problem. Many NP-hard optimization problems are now solved in practice using dynamic programming on low-treewidth graphs—see, e.g., [7, 1, 36]—so such a partition into bounded-treewidth graphs may also be practical. This decomposition approach has been successfully used to obtain constant-factor approximations for many graph problems, such as a 2-approximation for graph coloring in any odd-H-minor-free graph family [11] (a generalization of H-minor-free graphs), whereas on general graphs the problem is inapproximable within $n^{1-\varepsilon}$ for any $\varepsilon > 0$ unless ZPP = NP [19].

A generalization of this decomposition approach leads to PTASs for many minimization and maximization problems, such as vertex cover, minimum color sum, and hereditary problems such as independent set and max-clique [4, 18, 12]. The idea is to partition the vertices or edges of the graph into a small number k of pieces such that deleting any one of the pieces results in a bounded-treewidth graph (where the bound depends on k). Such a decomposition is known for planar graphs [4], bounded-genus graphs [18], apex-minor-free graphs [18], and H-minor-free graphs [15, 12]. However, this decomposition approach is effectively limited to problems whose optimal solution only improves when deleting edges or vertices from the graph. Bidimensionality theory [10] highlights *contracted-closed* problems, whose optimal solution only improves when contracting edges (but not necessarily when deleting edges), including classic problems such as dominating set (and its variations), minimum chordal completion, and the Traveling Salesman Problem (TSP).

These applications lead to the notion of *contraction decomposition*: partitioning the edges of a graph into a small number k of color classes such that contracting any one of the color classes results in a bounded-treewidth graph (where again the bound depends on k). Klein [26, 27] proved the first such result for planar graphs with a variation of contraction called compression (deletion in the dual graph). Demaine, Hajiaghayi, and Mohar [13] generalized contraction decomposition to graphs of bounded genus, or slightly more generally, "h-almost-embeddable" graphs.

The major open problem posed in that paper is generalizing this result to graphs excluding a fixed graph H as a minor, i.e., *H-minor-free graphs*. The seminal Robertson–Seymour Graph Minor decomposition theorem [32] states that all such graphs are clique-sums of h-almost-embeddable graphs. As described in [13], however, clique-sums are extremely difficult to work with in contraction decompositions because some of the edges in the join set of a clique sum are *virtual*: these edges are not in the actual graph, but appear in the individual (h-almost-embeddable) pieces. If we keep these virtual edges when applying the decomposition to each piece, the partition may assign some of these virtual edges to be contracted in certain cases, but the edges cannot actually be contracted because they do not exist in the actual graph. On the other hand, if we delete these virtual edges before applying the contraction decomposition, we still obtain that the pieces have bounded treewidth after contracting one of the classes, but it becomes impossible to join together these tree decompositions, because the join set no longer forms a clique and thus it is no longer contained in a single bag in each tree decomposition. A naïve combination of these tree decompositions causes a blowup in treewidth proportional to the number of clique-sum operations, which can be arbitrarily large. In contrast, this problem does not arise if we only delete edges within a color class as in [12], instead of contracting them, because the virtual edges can be deleted (indeed, they must be deleted, but this can only help), whereas they cannot be contracted. Indeed, as we show in this paper, new ideas are required to surmount these and other difficulties.

1.1 Our Results

The main result of this paper is that contraction decomposition is possible for H-minor-free graphs. This result is perhaps the ultimate in a series of contraction decompositions [26, 27, 13], and nicely parallels the deletion decomposition of H-minor-free graphs [15, 12].

THEOREM 1. *For any fixed graph H, there is a constant c_H such that, for any integer $k \geq 1$, every H-minor-free graph G can have its edges partitioned into $k+1$ color classes such that contracting any one of the color classes results in a graph of treewidth at most $c_H k$. Furthermore, such a partition can be found in polynomial time.*

Let us emphasize that the treewidth we obtain in Theorem 1 is linear in k, which is best possible dependence on k. This optimal dependence is important in our algorithmic applications below.

To show Theorem 1, we need several new insights and prove several new structural results. In particular, we strengthen the deep Graph Minor decomposition theorem of Robertson and Seymour by realizing the virtual edges involved in clique-sums by paths of real edges. This enables us to strengthen the induction hypothesis regarding edge partitions of h-almost-embeddable graphs, to avoid any blowup in the number of required paths beyond 3.

Along the way, in Section 3 we substantially strengthen and simplify contraction decompositions for bounded-genus graphs [13]. Specifically, we show that the partition can be computed by a simple radial breadth-first ball growth that is independent of both handles in the surface and special faces that we want to avoid contracting. Furthermore, we show that the breadth-first growth can start from a set of vertices instead of just one, as long as this set is *tight* in the sense that they induce a constant number of components of bounded treewidth.

We use our improved bounded-genus contraction decomposition to find three suitable paths to realize the virtual edges in a clique-sum, in the form of a $\bar{\tau}$ structure (pronounced "te structure"). We construct each of the three paths to follow an approximately shortest path in the radial graph. This idea of using radially approximately shortest paths is crucial because, as we show in Section 4, it implies that their union is tight in the sense above. This property allows us to obtain a good bound on treewidth using our improved bounded-genus contraction decomposition. To prove that a union of constantly many radially approximately shortest paths is tight, we introduce several new structural concepts such as dives and rainbows (Section 4.1), which seem interesting in their own right.

1.2 Algorithmic Applications

With the decomposition result in hand, we obtain the following general PTASs.

THEOREM 2. *Consider a minimization problem P on weighted graphs that is closed under contractions, solvable in polynomial time on graphs of bounded treewidth, and satisfying the following properties:*

1. *There is a polynomial-time algorithm that, given a weighted H-minor-free graph (G, w) and constant $\delta > 0$, computes an H-minor-free graph G' such that $\mathrm{OPT}(G') \geq \alpha \cdot w(G')$, for some constant $\alpha > 0$ (possibly depending on δ), and any c-approximate solution to G' can be converted into a $(1+\delta)\,c$-approximate solution to G in polynomial time. (G' is called a (δ, α)-*spanner *of G.)*

2. *There is a polynomial-time algorithm that, given a subset S of edges of a weighted graph (G, w), and given an optimal solution for G/S, constructs a solution for G of value at most $\mathrm{OPT}(G/S) + \beta\, w(S)$ for some constant $\beta > 0$.*

For any fixed minor H, and for any fixed $0 < \varepsilon \leq 1$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm for problem P in H-minor-free graphs. Furthermore, if α grows as a function of n, then the running time becomes bounded by a polynomial times the cost of solving the problem on graphs of treewidth $O(\alpha)$.

In particular Theorem 2 gives us PTASs for unweighted TSP and minimum-size c-edge-connected submultigraph[1], because every H-minor-free graph serves as its own unweighted spanner. TSP is a classic problem that has served as a testbed for almost every

[1]This problem allows using multiple copies of an edge in the input graph—hence sub*multi*graph—but the solution must pay for every copy.

new algorithmic idea over the past 50 years, and it has been considered extensively in planar graphs and its generalizations, starting with a PTAS for unweighted planar graphs [21] and a PTAS for weighted planar graphs [2] (since improved to linear time [26]).

In fact we can obtain a PTAS for TSP in *weighted* H-minor-free graphs, solving the main open problem posed in a seminal paper of Grohe [23]. For a weighted graph G excluding H as a minor (with $|V(H)| = h$), we have two properties:

1. for any $\varepsilon > 0$, Grigni and Sissokho [22] give a polynomial-time algorithm to find a spanning subgraph (i.e., a *spanner*) approximating all shortest-path distances within a factor of $1 + \varepsilon$, and with total edge weight at most $O((h\sqrt{\log h} \cdot \log n)/\varepsilon)$ times the weight of a minimum spanning tree; and

2. Dorn, Fomin, and Thilikos [16] show that, if G has treewidth at most k, then there is a $2^{f(h)\,k}n$-time algorithm to find an optimal-weight TSP for some function $f(h)$; see also [33].

Now we can apply Theorem 2 with $\alpha = O((h\sqrt{\log h}\cdot\log n)/\varepsilon) = O(\log n)$ for constant h and ε. Thus we obtain a polynomial-time $(1+\varepsilon)$-approximation for TSP in weighted H-minor-free graphs.

Theorem 1 also has several applications to obtaining fixed-parameter (exact) algorithms on H-minor-free graphs. For example, Kawarabayashi and Thorup [24] recently obtained a fixed-parameter algorithm with running time $O(2^k n)$ for the k-cut problem in planar graphs, where the goal is to remove a minimum number of edges in an undirected planar graph in order to form at least k connected components. This result solves the main open problem of Downey et al. [17]; in contrast, the problem is $W[1]$-hard (and thus unlikely to have such a fixed-parameter algorithm) for general graphs [20].

Using our contraction-decomposition theorem (Theorem 1), we show that it is easy to obtain a fixed-parameter algorithm for k-cut in graphs excluding a fixed minor H. Our proof is indeed an extension of Kawarabayashi and Thorup's proof for planar graphs. Because each H-minor-free graph has a vertex of degree at most $O(|V(H)|\sqrt{\lg|V(H)|})$ (see, e.g., [28, 34]), we know that the solution to the k-cut problem on H-minor-free graphs has size at most $c_h k$ for some constant c_h depending on $h = |H|$. Now if we use our contraction decomposition to partition the edges into $k = c_h k+1$ sets, at least one of the sets does not have any intersection with the optimum. Now by guessing this set among the $c_h k+1$ sets and contracting its edges, we are left with a graph of treewidth at most $O(k)$ in which we can solve the k-cut problem exactly in time $2^{\tilde{O}(k)}n$. Because the running time of our contraction-decomposition theorem is in $n^{O(1)}$ for fixed h, we obtain a fixed-parameter algorithm with overall running time $2^{\tilde{O}(k)}n + n^{O(1)}$.

We expect that several of the recent spanner results (as required by Theorem 2) for subset TSP [27], Steiner tree [9, 8], Steiner forest [5], and prize-collecting TSP and Steiner tree [6], for planar and bounded-genus graphs, extend to H-minor-free graphs as well. In this way, Theorem 2 will immediately result in PTASs for these problems in H-minor-free graphs.

2. OVERVIEW OF ALGORITHM

We now give an overview of our proof of Theorem 1. We first apply the seminal Graph Minor decomposition theorem to G, resulting in a clique-sum decomposition of G into h-almost-embeddable pieces. Recall that h-almost-embeddable graphs are bounded-genus graphs plus a bounded number of "vortices" and "apices"; refer to Appendix A for definitions.

As mentioned above, we can find a desired edge-coloring in each piece G_i. Thus, when we contract a single color class in each of the pieces, we obtain a tree decomposition of bounded width. Now suppose a piece G_i is clique-summed with a child piece in the clique-sum decomposition. In order to glue these two tree decompositions together, we need the following key additional property.

> **Key Property:** After contracting one color class, we obtain a tree decomposition W (of width at most $c_H k$) in the resulting graph, which has the property that the join set between any piece G_j and each child piece of G_j is contained in a single piece of W. Moreover, the apex vertex set A_j of any piece G_j is also contained in a single piece of W.

Note that the join set may involve either

1. two consecutive pieces of a vortex and apex vertex set A_j, or

2. at most three vertices in a face of a piece G_j and apex vertex set A_j.

In order to show the contraction-decomposition theorem with this property, we have to add some *virtual edges* in the bounded-genus part. Intuitively, vortices and apices are easy to handle when we construct the above tree decomposition W (we adapt the idea by Grohe [23] to deal with vortices and apices). However, in order to make sure that all the vertices (in the bounded-genus part) that are involved in a join set between some two pieces, are contained in a single piece of the tree decomposition W, we need to add virtual edges to the bounded-genus part.

Let us focus on one piece G_i of the clique-sum decomposition, and see what we shall do for virtual edges. So G_i is an h-almost-embeddable graph. If there is some child piece G' of G_i such that $G' \cap G_i$ involves the bounded-genus part of G_i, then we add all the missing edges in $G' \cap G_i$. By the seminal Graph Minor decomposition theorem, $|G' \cap G_i| \leq 3$, and the virtual edges can be embedded into the bounded-genus part of G_i. Hereafter, we assume that G_i contains all these virtual edges. These virtual edges indeed allow us to prove that G_i has a desired $(k+1)$-edge-coloring with the key property. This is done in Section 5, where we compute a $\bar{\tau}$ structure as described in Section 1.1.

Let us show how to proceed with our proof for the contraction-decomposition theorem. We have a rooted tree decomposition such that each piece is h-almost-embeddable graph. We begin with the root piece (which is the base case). It is not hard for the root piece to get a desired $(k+1)$-edge-coloring with the key property.

Suppose, inductively, that all the ancestor pieces of G_i plus G_i have a desired $(k+1)$-edge-coloring as in our contraction-decomposition theorem, with our key property.

Let a, b, c be three vertices of G_i that consist of a facial triangle in the bounded-genus part of G_i. Let $e_1 = ab$, $e_2 = bc$ and $e_3 = ca$. Let us observe that some of (or all of) e_1, e_2, e_3 may be virtual edges.

We now consider a child piece G_{i+1} of G_i. Ideally, we want to obtain the contraction-decomposition theorem for G_{i+1} with the above key property. Then after contracting one edge-color class, we would like to glue two tree decompositions W from all the ancestors of G_i (together with G_i), and W' from G_{i+1} together, to get a tree decomposition of bounded width. This is possible because by the key property, the join set between the piece G_i and each child piece of G_i is contained in a single piece of W, and moreover, the apex set A_{i+1} of G_{i+1} that contains the join set between G_i and G_{i+1}, is contained in a single piece of W'. Thus we can glue two tree decompositions W and W' together at the join set $G_i \cap G_{i+1}$.

However, there is an issue that the edges ab, bc, ca may be "virtual", i.e., some of ab, bc, ca may not be in the actual graph, but appear only in G_i. Thus if $G_i \cap G_{i+1}$ involves the edges ab, bc, ca, we have to make sure that those edges are actually contracted in G_{i+1} when we contract some of ab, bc, ca in G_i. In order to do that, we need to find at most three paths in G_{i+1} before we apply our contraction-decomposition theorem with the key property to G_{i+1}. More precisely, we shall find three paths P'_1, P'_2, P'_3 in G_{i+1} (in fact, these paths are allowed to go through descendant pieces of G_{i+1}) such that P'_1 connects a and b, P'_2 connects b and c, and P'_3 connects c and a, and P'_1, P'_2, P'_3 are edge-disjoint. [2]

Having found the paths P'_1, P'_2, P'_3 in G_{i+1}, we can make our most important point.

We shall precolor all the edges of P'_1 whose color is the same as that of ab in G_i. We do the same thing for P'_2, P'_3. Then our contraction-decomposition theorem has to be modified as follows:

> We have to allow the precoloring of P'_1, P'_2, P'_3. In other words, the conclusion of the contraction-decomposition theorem still holds under the condition that all the edges of P'_1, P'_2, P'_3 are precolored. In addition, it also has to satisfy the key property.

The proof of this modification diverges substantially from the arguments in [13]; see Section 3. Instead of handling handles in the surface and special faces separately at the end, we argue that a simple radial breadth-first coloring suffices, essentially replacing algorithmic complexity with analysis complexity. We also show that an entire set of vertices can serve as a root for the breadth-first search, provided that set is sufficiently "tight".

This precoloring guarantees us that when we contract the edge ab or bc or ca in G_i, although these edges may be virtual, they have to be contracted in G_{i+1} because we have to contract the corresponding path in P'_1, P'_2, P'_3 into a single point.

Thus if we could get a tree decomposition W' of G_{i+1} that satisfies the above assertion too (i.e, a $(k+1)$-edge-coloring of G_{i+1} with the key property and the above assertion), then we could glue two tree decompositions W and W' together at the join set $G_i \cap G_{i+1}$, to obtain one single tree decomposition of bounded treewidth (and the virtual edges are not a problem as we saw).

However, we have to make sure of the following:

> We only need to find at most three edge-disjoint paths in any children of G_{i+1}.

As far as we can see, there are two issues here.

The first issue is that the paths P'_1, P'_2, P'_3 may go into a child piece G' of G_{i+1} that is only attached to a vortex and the apices of G_{i+1} (let us call this child piece of G_{i+1} *type 1*). We want the paths P'_1, P'_2, P'_3 so that each goes into $G' - G_{i+1}$ at most once. In order to do that, we shall find the paths P'_1, P'_2, P'_3 in the bounded-genus part of G_{i+1} as much as possible. We shall prove that when the path P'_j first reaches the bounded-genus part of G_{i+1}, it never goes into any vortex, except possibly at the end. In addition, we shall prove that the bounded-genus part of all (but at most one *special* piece) of the child pieces of G_{i+1} of type 1 can hit at most two of the paths P'_1, P'_2, P'_3. Since these child pieces do not involve any virtual edge of G_{i+1}, thus we just need to find at most two edge-disjoint paths in the bounded-genus part of these child pieces of type 1 when we apply the inductive argument. In the special child piece of type 1, we may need to find three edge-disjoint paths when we apply our inductive argument. Therefore, we have to find at most three edge-disjoint paths in the child pieces of G_{i+1} of type 1.

There is one special case though. Some of the paths P'_1, P'_2, P'_3 may go through only vortices, and do not go into the bounded-genus part of any piece. In this case, we do not have any issue with virtual edges (because these paths do not pass through any virtual edges of any piece), and hence, we can just color the edges of these paths without creating any problem in the bounded-genus part of any piece.

The second issue is concerning some child pieces of G_{i+1} that involve the bounded-genus part of G_{i+1} (let us call these child pieces of G_{i+1} *type 2*). Since we have to add the virtual edges in the bounded-genus part of G_{i+1}, this means that we may need to find the corresponding (at most three) edge-disjoint paths in all the child pieces of G_{i+1} of type 2. On the other hand, if some of the paths P'_1, P'_2, P'_3 goes through some of the child pieces of G_{i+1} of type 2, we also need to find this path in some child piece G' of type 2, in addition to at most three edge-disjoint paths (that are needed because of the virtual edges in the bounded-genus part of G_{i+1}). This means that we may need to find not only at most three edge-disjoint paths in the child piece G' of G_{i+1}, but also one or more paths. This may be a big issue, because we may not be able to bound the number of edge-disjoint paths that we need to find in some piece of the clique-sum decomposition of H-minor-free graphs.

In order to resolve this problem, we shall prove the following.

> We shall prove that the above three edge-disjoint paths P'_1, P'_2, P'_3 can go into at most five *special* child pieces of G_{i+1}, among all the child pieces of G_{i+1} of type 2.
>
> In other words, all (but at most five special) child pieces of G_{i+1} of type 2 can only hit the paths P'_1, P'_2, P'_3 in the bounded-genus part of G_{i+1}. (hence these child pieces of G_{i+1} hit the paths P'_1, P'_2, P'_3 at their apices only, and in addition, the paths P'_1, P'_2, P'_3 do not contain any edge in these child pieces of G_{i+1}, except for some edges that are also present in G_{i+1}, including at most three virtual edges.).

In order to show this claim, intuitively, once some of the paths P'_1, P'_2, P'_3 go into a child piece of G_{i+1} of type 2, either we can find some of the paths P'_1, P'_2, P'_3 in this piece, or else some of the paths P'_1, P'_2, P'_3 go through this piece to the bounded-genus part of G_{i+1}. Moreover, in the second case, once the paths P'_1, P'_2, P'_3 reach the bounded-genus part of G_{i+1}, they never go into the child pieces of G_{i+1} of type 2, except possibly for the end of the paths. As mentioned above, we then find the paths P'_1, P'_2, P'_3 in the bounded-genus part of G_{i+1} as much as possible. In this way, we can show that there are at most five special child pieces of G_{i+1} as above.

We then delete the vertices that are both in one of these at most five special child pieces and in the bounded-genus part of G_{i+1}, from the bounded-genus part of G_{i+1} (thus we delete at most 15 vertices from the bounded-genus part of G_{i+1}), and put them to the apex vertex set A_{i+1}. Therefore, each of these at most five special child pieces is now only attached to the resulting apex vertex set A_{i+1} of G_{i+1}. This way, we are guaranteed that we only need to find at most three edge-disjoint paths in any child piece of the resulting graph of G_{i+1}, because in these special child pieces of G_{i+1}, there are no virtual edges of G_{i+1}, and other child pieces of G_{i+1} of type 2 would not give rise to any trouble, as claimed.

[2] In fact, we do not really require all three paths to be edge-disjoint. By overcoming one technical issue, we just requite that two of P'_1, P'_2, P'_3 are edge-disjoint. Intuitively, we manage to show that there is always a desired $(k+1)$-edge-coloring in G_i, satisfying the key property, such that two of e_1, e_2, e_3 receive the same color.

We have one more small issue. After the above modification of G_{i+1}, we need to keep the h-almost-embeddable structure in G_{i+1}. This is not hard, and will be taken care in Appendix B.

3. CONTRACTION DECOMPOSITION FOR BOUNDED-GENUS GRAPHS, IMPROVED

In this section, we develop both simpler and stronger forms of the contraction decomposition for bounded-genus graphs from [13]. The new coloring algorithm is simple, essentially just a breadth-first search. This makes the coloring oblivious to which faces are "special", and does not treat handles specially. This simplification complicates the analysis, but makes it easy to show that the algorithm has additional properties, in particular, that every face has at most two distinct colors (which we need later on). On the strengthening side, we show that the new coloring algorithm works when rooted at a more general set than just a single vertex (which we also need later on).

3.1 Radial Coloring

For a graph G 2-cell embedded in some surface, the *radial graph* $R = R(G)$ has a vertex for every vertex of G and for every face of G, and we label them the same: $V(R) = V(G) \cup F(G)$. $R(G)$ is bipartite with this bipartition. Two vertices $v \in V(G)$ and $f \in F(G)$ are adjacent in $R(G)$ if their corresponding vertex v and face f are incident. A *radial path* is a path in the radial graph. The *radial distance* between two vertices v, w in G (or $R(G)$) is the length of the shortest radial path between v and w. Define the radial distance between a vertex v and a vertex set S to be the minimum radial distance between v and a vertex in S.

The *radial coloring* from a root set R of vertices is defined as follows. For $i \geq 0$, define *vertex layer* i to consist of all vertices of G at radial distance $2i$ from R. (In particular, vertex layer 0 is R.) In other words, this layer decomposition can be seen as a breadth-first search in the radial graph (discarding the levels corresponding to faces) from the root vertices in R. For $i > 0$, define *face layer* i to consist of all faces of G at radial distance $2i - 1$ from R. For $i > 0$, define *edge layer* i to consist of all edges that first appear in face layer i, that is, they are edges of faces in face layer i but not faces in face layer $< i$. The radial coloring from root R defines *color class* i, for $1 \leq i \leq k$, to be the union of all edge layers $j \equiv i \pmod{k}$.

For $i > 0$, define *ball* i to be the union of the closure of face layers $1, 2, \ldots, i$; also define ball 0 to consist of the vertices in R. For $i \geq 0$, define *vertex boundary layer* i to consist of all vertices on the boundary of ball i, which is a subset of vertex layer i. Define *edge boundary layer* i to consist of all edges on the boundary of ball i, which is a subset of edge layer i. Thus, vertex boundary layer i and edge boundary layer i form a disjoint union of closed walks (being the boundary of closed regions).

Call a root R *(t, c)-tight*, for a monotone function t and integer c, if (1) the treewidth of the induced subgraph on the union of vertex layers $0, 1, \ldots, r$ is at most $t(r)$, and (2) the number of connected components in the induced subgraph $G[R]$ is at most c.

One simple case is when R is a single vertex; then R is $(t, 1)$-tight for a linear function t, by linear local treewidth in bounded-genus graphs [18]. Another example of a $(t, 1)$-tight root is when R consists of the vertices of a single face. Examples of $(t', O(1))$-tight roots are when R consists of a constant number k of vertices (or the vertices of a constant number k of faces), with the function $t'(r) = O(k) \cdot t(r)$.

LEMMA 3. *The radial coloring can be computed in polynomial time.*

LEMMA 4. *Any face has at most two different colors (and layer numbers) on its edges.*

We are now ready to prove the main theorem of this section: radial coloring in a bounded-genus graph from a tight root forms the desired contraction decomposition. Note that the coloring is oblivious to the choice of the q special faces.

THEOREM 5. *For every graph G of fixed (orientable) genus g, and for any integer $k \geq 2$, the radial coloring from a (t, c)-tight root R partitions the edges of G into k color classes such that contracting any one color class results in a graph of treewidth $O((c + g)k + t(k))$.*

Furthermore, if we mark as special *all edges among vertices in R and all edges of q faces of G, then the radial coloring from a (t, c)-tight root R partitions the nonspecial edges of G into k color classes such that contracting all (nonspecial) edges in one color class results in a graph of treewidth $O((c + g)qk + t(k))$.*

Proof: First we prove the first claim of the theorem (without special edges); later we describe the modifications necessary for the second claim (with special edges).

Consider the graph G' resulting from contracting one color class i, which consists of edge layers $i, i + k, i + 2k, \ldots$. In particular, contracting edge layer $i + jk$ contracts edge boundary layer $i + jk$ (which is a subset), each connected component of which is a closed walk. Thus, each closed walk in G contracts to a single point in G', called *articulation points*. Construct G'' by splitting each articulation point into two vertices, one connected to the neighbors in G with smaller layer numbers, and the other connected to the neighbors in G with larger layer numbers. We call the connected components of G'' *blobs*. Each blob consists of an interval of $k+1$ layer numbers $i + jk, \ldots, i + (j + 1)k$, where only the incident articulation points have the two extreme layer numbers. Define a directed acyclic graph B with a vertex for each blob, and one edge for each articulation point, connecting the two blobs with the two copies of the articulation point, with the edge directed from the layer interval of smaller numbers to the layer interval of larger numbers. A *root blob* is a blob containing a connected component of $G[R]$; these blobs correspond to the source vertices in B.

First we claim that the in-degree of each blob in B is at most $c + g$. When a blob has in-degree k, it corresponds to k frontiers of the breadth-first search (corresponding to the decontractions of the k articulation points) merging into one frontier. Initially we start with c frontiers, one per connected component of $G[R]$. Frontiers can split, but if split frontiers later remerge, we form a handle, and this can happen only g times. Thus the in-degree k is at most $c + g$.

Second we claim that each nonroot blob has radial diameter $O(k(g + c))$. By the breadth-first layer numbering, every vertex in the blob has radial distance at most $2k$ from the at most $c + g$ articulation points corresponding to the incoming edges in B. Thus we can partition the blob into at most $c + g$ chunks, where every vertex in chunk i is within radial distance $2k$ of the ith incoming articulation point. Hence chunk i has radial diameter at most $4k$: radial distance $2k$ to get from any vertex to the ith incoming articulation point, and radial distance $2k$ to get from there to any other vertex in the chunk. By definition, the blob is connected, so the chunks are connected together by edges in the blob. In the worst case, these connections from a path, resulting in an overall blob diameter of at most $(4k + 1)(c + g)$.

Third we claim that we can remove g edges in B to remove all cycles from B (ignoring edge orientations). Essentially we just remove one edge per handle.

Define G''' by starting from G' and splitting into two just the g articulation points necessary to make B acyclic (splitting in the same way as for G''). Because the new blob graph B' is acyclic, G''' can be written as a tree of clique 1-sums of its constituent blobs. The radial diameter of each nonroot blob is $O(k(c+g))$ by the second claim. By Eppstein [18], the treewidth of each nonroot blob is proportional to its radial diameter, so is also $O(k(c+g))$. By tightness, the treewidth of the union of the root blobs is $t(i) \leq t(k)$. The treewidth of a clique-sum is the maximum of the treewidths of the terms [14], so the treewidth of G''' is $O(k(c+g)+t(k))$.

Given a tree decomposition of G''', we can modify it into a tree decomposition of G' by adding, to all bags in the tree decomposition, each of the g articulation points that we split. (Also we remove the two copies of these articulation points from the bags in which they appear.) This modification increases the treewidth by an additive g, so the bound remains $O(k(c+g)+t(k))$.

Finally we turn to the second claim of the theorem, with special edges. Marking the edges among vertices in R as special only prevents some edge contractions within the root blobs, because they all have layer number 0. But the argument above required no contractions to happen within the root blobs, so the root blobs still have bounded treewidth as desired. Marking the edges of q faces in G as special will prevent some closed walks of edge layers $i+jk$ from contracting to single articulation points. By Lemma 4, each of the q faces lies in at most two edge layers; in fact, the two edge layer numbers are consecutive, so only one will be of the form $i+jk$ (because $k \geq 2$). Let J be the set of j values for which edge layer $i+jk$ contains a special edges. Because $|J| \leq q$, there exists an integer $x \in \{0,1,\ldots,q\}$ for which $q \not\equiv j \pmod{q+1}$ for all $j \in J$. Now consider edge layers $i+k(x+j(q+1))$ for $j=0,1,\ldots$, which by construction contain no special edges. These layers form a subset of the edge layers $i+jk$, with a regular spacing of $k(q+1)$ instead of k. Thus we can apply the arguments above but with k replaced by $k(q+1)$ (and i replaced by $i+kx$) and obtain a treewidth bound of $O(k(q+1)(c+g)+t(k))$. □

4. SHORTEST PATHS ARE TIGHT

In this section, we prove that the union of a constant number of radial shortest paths is a valid root for radial coloring:

THEOREM 6. *The vertices visited by c radial shortest paths in a bounded-genus graph G is (t,c)-tight, where $t(r)=O(r)$. The same result holds if each radial path is within an additive constant of shortest.*

To prove this theorem, we need to show that a radius-r radial neighborhood of the paths has small treewidth, or equivalently, has no large wall. The proof is by contradiction: if we had a large wall, then the paths must travel deep into the wall (called a "dive"), which translates into a series of possible shortcuts for the path (called a "rainbow"), which eventually causes a contradiction. The new concepts of dives and rainbows may be of independent interest, and we turn to them now.

4.1 Dives and Rainbows

We define an "r-dive" as follows. Suppose H is a planar graph that contains an $2r$-wall. Let $C_1,\ldots,C_r$ be vertex-disjoint cycles in the plane graph H. Let D_i be the disc in the plane with boundary C_i. We say that they are *concentric* if we have the property that $D_r \subseteq D_{r-1} \subseteq \cdots \subseteq D_1$. Consider the radial graph of H. An *r-dive* is a radial path within the disc D_1, with both endpoints in C_1, and with at least one vertex in C_k for some $k \geq r$.

Given a radial path P, an *r-rainbow* on P consists of r vertex-disjoint paths $P_1,P_2,\ldots,P_r$ in H, all on the same side of P, where x_i, y_i are the endpoints of P_i (and thus being vertices, not faces, of H), so that $x_r,x_{r-1},\ldots,x_1,y_1,y_2,\ldots,y_r$ appear in this order along P.

LEMMA 7. *Given an r-dive P, we can construct an r-rainbow on P.*

Proof: We construct r paths $P_1,P_2,\ldots,P_r$, forming an r-rainbow, using subpaths of the cycles $C_1,C_2,\ldots,C_r$, respectively.

To do so, we map the dive to a subscripted balanced-parenthesis expression by following along the path P, writing $(_i$ each time the dive enters D_i, and writing $)_i$ each time the dive exits D_i. Each instance of $(_i$ or $)_i$ corresponds to a vertex of C_i where the path P enters or exits D_i. By planarity of the containing graph H and thus the path P, the subscripted parenthesis string is indeed balanced (with matching subscripts). Furthermore, by nesting, planarity, and vertex-disjointness of the cycles, the parent pair $(_i\cdots)_i$ immediately containing a child pair $(_j\cdots)_j$ must have $j=i+1$ (levels cannot be skipped).

Now we take the parenthesis pair $(_q\cdots)_q$ with maximum subscript q, as well as its parent pair $(_{q-1}\cdots)_{q-1}$, grandparent pair $(_{q-2}\cdots)_{q-2}$, etc., to its outermost ancestor $(_1\cdots)_1$. We obtain a balanced-parenthesis substructure $(_1\cdots(_2\cdots(_q\cdots)_q\cdots)_2\cdots)_1$. Each pair $(_i\cdots)_i$ defines two interior-disjoint paths along C_i between the corresponding entrance and exit points on C_i. We choose P_i among these two paths consistently for all i, say, always connecting from the entrance on the right side of P to the exit on the left side of P (where left and right are defined by planarity and an arbitrary orientation of P).

Finally we show that these paths $P_1,P_2,\ldots,P_r$ form an r-rainbow on P. We have constructed the P_i's to lie all on the same side of P. Disjointness of the P_i's follows because C_i's are vertex-disjoint, and each P_i is a subpath of C_i. The desired order of endpoints follows from the nesting of the C_i's. □

LEMMA 8. *Let P be a radial shortest path in the surface, and suppose it has an (cr)-rainbow for $c>16$. Then there is a wall W of size at least $\frac{1}{4}cr$ in the planar graph Q bounded by $x_{cr}, P, x_{\frac{1}{2}cr+1}, P_{\frac{1}{2}cr}, y_{\frac{1}{2}cr+1}, P, y_{cr}, P_{cr}$ such that the radial distance between W and P is more than $2r$. Note that this region is on the same side of P as the rainbow paths $P_1,P_2,\ldots,P_{cr}$.*

Proof: Because subpaths of (radial) shortest paths are (radial) shortest paths, a subpath of P is a radial shortest path from any x_i to any y_j. In particular, vertices $x_{i-1},x_{i-2},\ldots,x_1,y_1,y_2,\ldots,y_{j-1}$ appear along that subpath. Because the x_i's and y_j's are all distinct (by vertex-disjointness of the rainbow paths), and must be intervened by faces in the radial subpath of P, the radial distance between x_i and y_j must be at least $2(i+j-1)$.

We claim that there are at least cr vertex-disjoint paths in Q between P_{cr} and $P_{\frac{1}{2}cr+1}$. For otherwise, by Menger's Theorem, there is a radial path C across Q, separating P_{cr} from $P_{\frac{1}{2}cr+1}$, that hits at most $cr-1$ vertices in Q. Because Q is bounded by four sides—P_{cr}, $P_{\frac{1}{2}cr+1}$, and two subpaths of P—while the radial path C separates the first two sides, the endpoints of C must lie along P, say between x_i and x_{i+1} (inclusive) and between y_j and y_{j+1} (inclusive), respectively. Thus the radial path C offers a potential shortcut for P. The (radial) length of C is at most $2[cr-1]$, yet by the argument above we know that the radial distance between the endpoints is at least $2(i+j-1) \geq 2[cr+1]$ (because $i, j \geq \frac{1}{2}cr+1$). This contradicts the assumption that P is a radially shortest path, proving the claim.

Now we combine the $\frac{1}{2}cr$ vertex-disjoint paths $P_{\frac{1}{2}cr+1}, P_{\frac{1}{2}cr+2}, \ldots, P_{cr}$ with the cr vertex-disjoint paths between $P_{\frac{1}{2}cr+1}$ and P_{cr} (from the previous claim) to form a subdivided $\frac{1}{2}cr \times cr$ grid in Q. By dropping alternating portions of the cr vertex-disjoint paths, we form a $\frac{1}{2}cr \times \frac{1}{2}cr$ wall in Q. Finally we pick the central $\frac{1}{4}cr \times \frac{1}{4}cr$ subwall W of this wall, so that the distance between W and P is at least $\frac{1}{8}cr > 2r$ for $c > 16$. This proves the lemma. □

4.2 Tightness of Shortest Paths

Proof of Theorem 6: First we consider the case that all c paths are radially shortest; at the end we will extend to the approximately shortest case. Let R be the union of the vertices in the c given radial shortest paths, which clearly induces at most c connected components. Let G' denote the induced subgraph of G on the union of vertex layers $0, 1, \ldots, r$ in the radial coloring. It remains to show that G' has treewidth $O(r)$.

If G' has treewidth w, then it has a wall of size $\Omega(w)$. In fact, because G and hence G' has genus at most g, G' must have a *planar wall* of size $\Omega(w/\sqrt{g})$: all parts of the graph G' contained within the outer boundary of the wall form a planar graph. This result was proved by Mohar [31] and Thomassen [35, Proposition 3.1].

We can modify this planar wall into another planar wall of size $\Omega(w/\sqrt{g})$ with the property that all the endpoints of the shortest paths, and radius-$2r$ radial neighborhoods around them, are outside the wall. Namely, annihilate from the wall the r rows and columns above, below, left, and right of each such endpoint, and choose the largest subwall that remains. Each of these annihilations loses an additive $2r+1$ rows and columns and then a multiplicative factor of 2 in the wall size. Repeating for each of the $2c$ endpoints, if we started with a wall of size $d2^{2c}4r$, then we will still have a wall of size $d\,r$. Because every point in G' and hence the wall is within radial distance $2r$ of a vertex on one of the c shortest paths, but we removed from the wall the radius-$2r$ radial neighborhoods of the $2c$ endpoints of these paths, we now have every vertex of the wall within radial distance $2r$ of a non-endpoint vertex of one of the c shortest paths.

The contents of the outside face of the planar wall of size $d\,r$ determine a planar graph H with $\frac{1}{2}d\,r$ concentric vertex-disjoint cycles. Consider a shortest path P_1 that comes within radial distance $2r$ of the center vertex of the wall. Because this path has both endpoints outside H, it forms a $(\frac{1}{2}d-1)r$-dive in H. By Lemma 7, we obtain a $(\frac{1}{2}d-1)r$-rainbow on the path P_i. By Lemma 8, we obtain a wall W_1 far from P_1 with treewidth $\Omega(r)$. We again know that some path, call it P_2, must dive deep into the wall W_1 in order for those vertices to be covered by the radius-k neighborhood of the paths. Thus we can repeat this process (find nested cycles in the wall, construct a dive, construct a rainbow, and find a safe region) for P_2 constrained within W_1. (We do not repeat the initial constant-factor culling necessary to get the endpoints outside of the wall; we just do that once at the beginning.) This results in a smaller wall W_2 far from both P_1 (being a subgraph of W_1) and P_2. Then we repeat for $P_3, P_4, \ldots, P_c$ constrained to previous walls. For each of these steps, we lose a multiplicative $\frac{1}{4}$ in the size of the wall. Set d sufficiently large so that the final wall W_c has positive size. Thus we get a vertex (in W_c) that is far from all c shortest paths. But this contradicts that we are in G'.

Finally we describe the necessary changes to this argument for the approximately shortest case. Given a radial path that is within an additive a of shortest, every subpath is also within an additive a of shortest; thus we still have the hereditary property crucial to the proof of Lemma 8. By trimming the wall in Lemma 8 smaller by an additive $O(a)$, we can not only find shortcuts to a radially shortest path, but find a shortcut that is more than a shorter than a given path, and thus contradict being within an additive a of shortest. Therefore the theorem holds. □

5. KEY THEOREM

Given five vertices a_1, b_1, a_2, b_2, c_1 in the bounded-genus part $\tilde{G}$ of an h-almost-embeddable graph G, define a $\bar{\tau}$ *structure* (pronounced "te structure") on a_1, b_1, a_2, b_2, c_1 to be three internally edge disjoint paths P_1, P_2, P_3 in $\tilde{G}$, where P_1 is a path from a_1 to b_1, P_2 is a path from a_2 to b_2, and P_3 is a path from c_1 to a vertex of P_1. In fact, we will be interested in $\bar{\tau}$ structures when a_1 and a_2 are neighbors of a common apex a, and similarly b_1 and b_2 are neighbors of a common apex b, and c_1 is a neighbor of an apex c, in order to form paths that replace virtual edges among apices a, b, c.

With a slightly abuse of the notation, sometimes the $\bar{\tau}$ structure with respect to a_1, a_2, b_1, b_2, c means that all the vertices of a_1, a_2, b_1, b_2, c are in the bounded-genus part $\tilde{G}$ of G, and there are edge-disjoint paths P_1, P_2, P_3 in $\tilde{G}$, such that P_1 and P_2 are paths joining a_1, a_2 and b_1, b_2, and P_3 is a path from c to a vertex of P_1 other than a.

THEOREM 9 (KEY THEOREM). *Suppose we are given an h-almost-embeddable graph G and five vertices a_1, a_2, b_1, b_2, c_1 in the bounded-genus part of G. Then we can compute a $\bar{\tau}$ structure P_1, P_2, P_3 on a_1, a_2, b_1, b_2, c_1 and a partition of G's edges into k color classes plus one class of special (colorless) edges such that (1) all edges in $P_1 \cup P_2 \cup P_3$, and all edges between pairs of apices, are special; and (2) contracting all (nonspecial) edges in one color class results in a graph with a tree decomposition W having the following properties:*

- *(a) the width of W is at most $f(k, h)$,*
- *(b) every apex of G is in every bag of W,*
- *(c) any two consecutive bags of a vortex appear in a common bag in W, and*
- *(d) every triangle in the bounded-genus part of G has its edges colored by only one or two colors.*

Proof: We construct a "shortest" $\bar{\tau}$ structure P_1, P_2, P_3 on a_1, a_2, b_1, b_2, c_1 in the bounded-genus part $\tilde{G}$ (where all apices and vortices have been removed) as follows. First we compute a shortest radial path R_1 from a_1 to b_1 and a shortest radial path R_2 from a_2 to b_2. If paths R_1 and R_2 properly cross,[3] we can uncross them into paths R_1' and R_2' by rerouting each crossing such that $|R_1| + |R_2| = |R_1'| + |R_2'|$. If R_1 and R_2 have an even number of proper crossings, then R_1' is a radial path from a_1 to b_1 and R_2' is a radial path from a_2 to b_2. If R_1 and R_2 have an odd number of proper crossings, then R_1' is a radial path from a_1 to b_2 and R_2' is a radial path from a_2 to b_1. Let R_1'' be a shortest radial path between the endpoints of R_1', and let R_2'' be a shortest radial path between the endpoints of R_2'. Clearly $|R_1''| \leq |R_1'|$ and $|R_2''| \leq |R_2'|$. In the case of an even number of proper crossings, we can let $R_1'' = R_1$ and $R_2'' = R_2$, so we also have $|R_1''| + |R_2''| = |R_1'| + |R_2'|$. If $|R_1''| + |R_2''| = |R_1'| + |R_2'|$, then $|R_1''| = |R_1'|$ and $|R_1''| = |R_1'|$, so in fact R_1' and R_2' are shortest radial paths between their endpoints. In the case of an odd number of proper crossings, we

[3]Radial paths R_1 and R_2 *properly cross* if they visit a common face f with $R_1 = \ldots, v_1, f, w_1, \ldots$ and $R_2 = \ldots, v_2, f, w_2, \ldots$ and v_1, v_2, w_1, w_2 appear in cyclic order around face f.

can have $|R_1''| + |R_2''| < |R_1'| + |R_2'|$. If R_1'' and R_2'' properly cross, we again perform uncrossing. If R_1'' and R_2'' have an even number of proper crossings, we obtain radial paths R_1''' and R_2''' with the same endpoints as R_1'' and R_2'', respectively, such that $|R_1'''| + |R_2'''| = |R_1''| + |R_2''|$, $|R_1''| \leq |R_1'''|$, and $|R_2''| \leq |R_2'''$, so $|R_1''| = |R_1'''|$ and $|R_2''| = |R_2''|$. Thus R_1''' and R_3''' are also shortest radial paths between their endpoints. If R_1'' and R_2'' have an odd number of proper crossings, we obtain radial paths R_1''' and R_2''' with the same endpoints as R_1 and R_2, respectively, such that $|R_1'''| + |R_2'''| = |R_1''| + |R_2''|$. Because R_1 and R_2 were the shortest such paths, we have $|R_1| \leq |R_1'''|$ and $|R_2| \leq |R_2'''|$. But then $|R_1'''| + |R_2'''| = |R_1''| + |R_2''| < |R_1'| + |R_2'| = |R_1| + |R_2| \leq |R_1'''| + |R_2'''|$, a contradiction.

Thus, in all cases, by possibly swapping the labels of b_1 and b_2, we can find a shortest radial path R_1 from a_1 to b_1 and a shortest radial path R_2 from a_2 to b_2 such that R_1 and R_2 do not properly cross.

Next, we define R_3 to be the shortest radial path from c_1 to any vertex on either R_1 or R_2 (including their endpoints a_1, b_1, a_2, b_2).

We construct paths P_1, P_2, P_3 in G by walking around the radial paths R_1, R_2, R_3. For $i \in \{1, 2\}$, there are two possible paths $P_i = a_i, \ldots, b_i$ in $\tilde{G}$ that walk around radial path R_i, visiting each vertex of G in R_i and walking around either the left or right side of each face of G in R_i. Because R_i is a shortest radial path, it repeats no faces, so the resulting paths repeat no edges. If R_1 and R_2 touch, then they touch on only one side, and we choose P_1 and P_2 to traverse the other sides, guaranteeing edge disjointness. If R_1 and R_2 are disjoint, we choose the sides for P_1 and P_2 arbitrarily. Next, we choose P_3 to walk around R_3 on any side, and once it reaches a face shared by R_1 and R_3, we continue P_3 around that face until it reaches a vertex of P_1 or P_2. By suitable swapping of labels, we can assume that P_3 ends at a vertex of P_1, which implies that R_3's endpoint other than c_1 is on R_1.

Thus we have paths P_1, P_2, and P_3 forming a $\bar{\tau}$ structure such that each path P_i visits the same faces as the corresponding shortest radial paths R_i.

Now we modify the bounded-genus part $\tilde{G}$ to "represent" each vortex by a cycle (similar in spirit to Grohe [23, Proposition 10]). More precisely, for a vortex attached to $\tilde{G}$ at vertices $v_0, v_1, \ldots, v_{k-1}$ in order around a face of $\tilde{G}$, we add edges $\{v_i, v_{(i+1) \bmod k}\}$ for $i \in \{0, 1, \ldots, k-1\}$, forming a new face. Let G' denote this modification of $\tilde{G}$ with a new face for each vortex, and mark these new faces special.

This change involves addition of edges, so cannot decrease any radial distance. Furthermore, the modification increases the radial distance by at most an additive 4 per vortex, replacing a visit of the face containing $v_0, v_1, \ldots, v_{k-1}$ with a visit of at most three faces in G'. In this way, we can modify radial paths R_1, R_2, R_3 in $\tilde{G}$ to form radial paths R_1', R_2, R_3 in G' to be still approximately shortest, within an additive $4h$ of the shortest possible length.

By Theorem 6, $R_1' \cup R_2' \cup R_3'$ is $(t, 2)$-tight in G' for $t(r) = O(r)$. Because path P_i is always within radial distance $O(1)$ of R_i', we also have that $P_1 \cup P_2 \cup P_3$ is $(t', 2)$-tight in G' for $t'(r) = O(r)$.

Now we apply Theorem 5, with $P_1 \cup P_2 \cup P_3$ as the root for the radial coloring in G', and with the previously mentioned vortex faces marked special. Thus contracting G' along all nonspecial edges in one color class results in a graph with a tree decomposition of width $O((h + h)hk + t(k)) = O(h^2 k)$. By Lemma 4, the coloring satisfies Property (d).

Given such a partition of the edges of G', we can extend the partition to the original graph G (with vortices and apices), by making all edges in G but not G' special. Now if we contract the edges in one color class in G', we obtain a tree decomposition with Properties (a), (b), and (d). We can extend this tree decomposition to the corresponding contraction of G (with the same color class of edges contracted) by adding all apices to all bags, thus satisfying Property (b), and replacing each occurrence of v_i in the tree decomposition with the entire bag from the path decomposition of the vortex. It is easy to see that Properties (a), (b), and (d) are preserved. We also obtain Property (c) because $\{v_i, v_{i+1}\}$ is an edge in G', so both v_i and v_{i+1} appear in a common bag in the tree decomposition for G', and thus the corresponding bags of the vortex appear in a common bag in the tree decomposition for G.

The width of the decomposition increases by a factor of $O(h)$: we lose an additive h from adding each apex to each bag, and lose at most a multiplicative h from exploding each v_i into a bag of size at most h. Thus we satisfy Property (a). □

6. REFERENCES

[1] E. 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.

[2] S. Arora, M. Grigni, D. Karger, P. Klein, and A. Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 33–41, 1998.

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

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

[5] M. Bateni, M. Hajiaghayi, and D. Marx. Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth. In *Proceedings of the 42nd ACM Symposium on Theory of computing (STOC)*, pages 211–220. ACM, 2010. Journal version submitted to J. ACM.

[6] M. H. Bateni, M. T. Hajiaghayi, and D. Marx. Prize-collecting network design on planar graphs. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, 2010. To appear.

[7] H. 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.

[8] G. Borradaile, E. D. Demaine, and S. Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science*, pages 171–182, Freiburg, Germany, February 2009.

[9] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[10] E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2008.

[11] E. D. Demaine, M. Hajiaghayi, and K. ichi Kawarabayashi. Decomposition, approximation, and coloring of

odd-minor-free graphs. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 329–344, Austin, Texas, January 2010.

[12] E. D. Demaine, M. Hajiaghayi, and K. 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.

[13] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*. To appear. Previously appeared at SODA 2007.

[14] 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(2):166–195, September 2004.

[15] 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(1):25–41, 2004.

[16] F. Dorn, F. Fomin, and D. Thilikos. Catalan structures and dynamic programming in h-minor-free graphs. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 631–640, 2008.

[17] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto, and F. Rosamond. Cutting up is hard to do: the parameterized complexity of k-cut and related problems. *Electr. Notes Theor. Comput. Sci.*, 78, 2003.

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

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

[20] M. R. Fellows. New directions and new challenges in algorithm design and complexity, parameterized. In *Proceedings of the 8th International Workshop on Algorithms and Data Structures*, volume 2748 of *Lecture Notes in Computer Science*, pages 505–520, Ottawa, Ontario, Canada, July–August 2003.

[21] M. Grigni, E. Koutsoupias, and C. Papadimitriou. An approximation scheme for planar graph TSP. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (Milwaukee, WI, 1995)*, pages 640–645, Los Alamitos, CA, 1995.

[22] M. Grigni and P. Sissokho. Light spanners and approximate TSP in weighted graphs with forbidden minors. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 852–857, 2002.

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

[24] K.-i. Kawarabayashi and M. Thorup. Minimum k-way cut of bounded size is fixed-parameter tractable. Preprint.

[25] K.-i. Kawarabayashi and P. Wollan. A simpler algorithm and shorter proof for the graph minor decomposition. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*, 2010.

[26] P. 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.

[27] P. N. Klein. A subset spanner for planar graphs, with application to subset TSP. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, pages 749–756, 2006.

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

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

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

[31] B. Mohar. Combinatorial local planarity and the width of graph embeddings. *Canadian Journal of Mathematics*, 44(6):1272–1288, 1992.

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

[33] S. Tazari. Faster approximation schemes and parameterized algorithms on h-minor-free and odd-minor-free graphs. In *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*, volume 6281 of *Lecture Notes in Computer Science*, pages 641–652, 2010.

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

[35] C. Thomassen. A simpler proof of the excluded minor theorem for higher surfaces. *Journal of Combinatorial Theory, Series B*, 70:306–311, 1997.

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

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

APPENDIX

A. GRAPH MINOR DECOMPOSITION

This section describes the Robertson-Seymour decomposition theorem characterizing the structure of H-minor-free graphs and the relevant basic concepts. There are some overlaps in the previous section, but for reader's convenience, we present all notations and definitions here.

A *separation* (A, B) is such that $G = A \cup B$, $A - B \neq \emptyset$, $B - A \neq \emptyset$, and there are no edges between $A - B$ and $B - A$. The order of the separation (A, B) is $|A \cap B|$.

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

Second, we need a basic notion of embedding.

A *surface* Σ is a compact 2-manifold, without boundary. 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 *line* in Σ is subset homeomorphic to $[0, 1]$. An O-arc is a subset of Σ homeomorphic to a circle. Let G be a graph 2-cell embedded in Σ, i.e., every region in the embedding is homeomorphic to a disc. To simplify notations we do not distinguish between a vertex of G and the point of Σ used in the drawing to represent the vertex or between an edge and the line representing it. We also consider G as the union of the points corresponding to its vertices and edges. That way, a subgraph H of G can be seen as a graph H where $H \subseteq G$. We call by *region* of G any connected component of $\Sigma - E(G) - V(G)$. (Every region is an open set.) We use the notation $V(G)$, $E(G)$, and $R(G)$ for the set of the vertices, edges and regions of G.

If $\Delta \subseteq \Sigma$, then $\overline{\Delta}$ denotes the *closure* of Δ, and the boundary of Δ is $\mathrm{bor}(\Delta) = \overline{\Delta} \cap \overline{\Sigma - \Delta}$. An edge e (a vertex v) is incident with a region r if $e \subseteq \mathrm{bor}(r)$ ($v \in \mathrm{bor}(r)$).

A subset of Σ meeting the drawing only in vertices of G is called *G-normal*. If an O-arc is G-normal then we call it *noose*. The length of a noose is the number of its vertices. $\Delta \subseteq \Sigma$ is an open disc if it is homeomorphic to $\{(x, y) : x^2 + y^2 < 1\}$. We say that a disc D is *bounded* by a noose N if $N = \mathrm{bor}(D)$. A graph G 2-cell embedded in a connected surface Σ is *θ-representative* if every noose of length $< \theta$ is contractable (null-homotopic in Σ).

At a high level, the deep decomposition theorem of Robertson and Seymour [32, 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* or *the apex vertex set*) 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 $U_i := V(G_0) \cap V(G_i) = V(G_0) \cap D_i$ (the vertices in U_i are sometimes called *society vertices* and the disks $D_1, \ldots, D_h$ are sometimes called *cuffs*); and
4. the graph G_i has a path decomposition $(B_u)_{u \in U_i}$ of width less than h, such that $u \in B_u$ for all $u \in U_i$. The sets B_u are ordered by the ordering of their indices u as points along the boundary cycle of face F_i in G_0.

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 [37] 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 $h : W_1 \rightarrow W_2$. 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*. We sometime call the join set *virtual clique* because of the above construction. 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 10. *[32, Theorem 1.3] For every graph H, there exists an integer $h \geq 0$ depending only on $|V(H)|$ such that every H-minor-free graph can be obtained by at most h-sums of graphs that are h-almost-embeddable in some surfaces in which H cannot be embedded.*

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

A polynomial-time algorithm for computing the structure guaranteed by this theorem is obtained in [12]. Recently, an easier $O(n^3)$ algorithm is found in [25].

One of the most important results concerning the treewidth is existence of grid minor or a wall. An *r-wall* is a graph which is isomorphic to a subdivision of the graph W_r with vertex set $V(W_r) = \{(i, j) \mid 1 \leq i \leq r, 1 \leq j \leq r\}$ in which two vertices (i, j) and (i', j') are adjacent if and only if one of the following possibilities holds:

(1) $i' = i$ and $j' \in \{j - 1, j + 1\}$.

(2) $j' = j$ and $i' = i + (-1)^{i+j}$.

We can also define an $(a \times b)$-wall in a natural way. It is easy to see that if G has an $(a \times b)$-wall, then it has an $(a \times b)$-grid minor, and conversely, if G has an $(a \times b)$-grid minor, then it has an $(a/2 \times b)$-wall. Let us recall that the $(a \times b)$-grid is the Cartesian product of paths $P_a \square P_b$. The (4×5)-grid and the (8×5)-wall are shown in Figure 1.

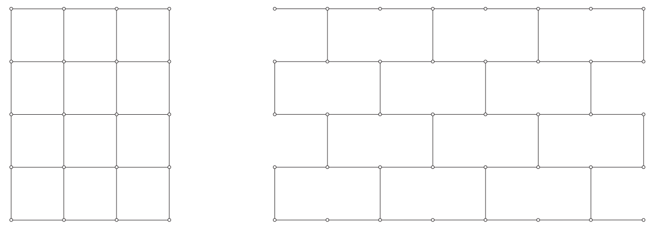

Figure 1: The (4×5)-grid and the (8×5)-wall

Finally, let us define canonical cycles $C_1, \ldots, C_k$ in a $2k$-wall W. We now delete vertices of degree 1 in W. Then C_k is an outer face boundary. Inductively, we can define C_i, which is the outer face boundary obtained by W by deleting all the cycles $C_k, \ldots, C_{i+1}$.

Let H be an r-wall in G. If G is embedded in a surface S, then we say that the wall H is *flat* if the outer cycle of H bounds a disk in S and H is contained in this disk. The following theorem was proved by Thomassen [35].

THEOREM 11. *For every r and g, if a graph G is embedded in a surface of Euler genus at most g and has treewidth at least $6rg^3$, then G contains a flat r-wall. Hence, if there is no flat r-wall, then the treewidth of G is at most $6rg^3$.*

B. MODIFYING THE CLIQUE-SUM DECOMPOSITION

We use the following strengthened forms of Theorem 10:

THEOREM 12. *The clique-sum decomposition of Theorem 10, 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 apex vertex set A_i in G_i. Furthermore, the join set between the piece G_i and its child piece G_{i+1} contains at most three vertices from the bounded-genus part of G_i (where the bounded-genus part of G_i is obtained from G_i by excluding A_i and all the vortices of G_i, but including all the society vertices of the vortices). Moreover,*

(I) if G_{i+1} involves the bounded-genus part of G_i, then it is properly attached to G_i, and

(II) for a fixed constant l, if we replace h in the definition of clique-sum decomposition by $f(h, l)$ (for some function f of h, l), then we can make the representativity of the bounded-genus part of G_i (for all i) at least l.

THEOREM 13. *The clique-sum decomposition of Theorem 12, written as $G_1 \oplus G_2 \oplus \cdots \oplus G_k$, has the following additional properties:*

> *Let G_i be a piece. If we take at most five special child pieces of G_i that involve the bounded-genus part of G_i, and*
>
> 1. *we delete all the vertices that are contained both in one of these at most five special child pieces and in the bounded-genus part of G_i, and then*
> 2. *we put them to the apex vertex set A_i of G_i,*
>
> *then we still have the structure as in Theorem 12 (with l replaced by $l - 15$ in (II)).*

Note that we would put at most 15 vertices to the apex vertex set A_i of G_i, since each child piece of G_i contains at most three vertices that are in the bounded-genus part of G_i.