Contractions of graphs on surfaces in polynomial time*

Marcin Kamiński[†] Daniël Paulusma[‡]
Dimitrios M. Thilikos[§]

December 20, 2010

Abstract

We prove that for every integer $g \geq 0$ and graph H, there exists a polynomial-time algorithm deciding whether an input graph of Euler genus at most g can be contracted to H.

We introduce surface contractions and surface topological minors of embedded graphs. We prove that an embedded graph H is a surface contraction of an embedded graph G if and only if the geometric dual of H is a surface topological minor of the geometric dual of G.

Keywords: graphs on surfaces, geometric dual, contraction, topological minor

*This research was done while the two last authors were visiting the Département d'Informatique Université Libre de Bruxelles in January 2010. A preliminary version of this work appeared in the Proceedings of the 18th Annual European Symposium on Algorithms ESA 2010 [11].

[†]The author is Chargé de Recherches du F.R.S. – FNRS; Département d'Informatique, Université Libre de Bruxelles, Belgium, Marcin.Kaminski@ulb.ac.be

[‡]School of Engineering and Computing Sciences, University of Durham, United Kingdom, supported by EPSRC (EP/G043434/1), Daniel.Paulusma@durham.ac.uk

[§]Departement of Mathematics, National and Kapodistrian University of Athens, Greece, sedthilk@math.uoa.gr, supported by the project "Kapodistrias" (AΠ 02839/28.07.2008) of the National and Kapodistrian University of Athens

1 Introduction

An *edge contraction* of an edge e in a graph is the graph obtained by removing e, identifying its two endpoints, and eliminating parallel edges that may appear. Some basic properties of contractions are collected in [23]. Formally, for an edge e with endpoints u and w, the contraction of e is the graph with vertex set $V(G/e) = V(G) \setminus \{u, w\} \cup \{v_{uw}\}$ and edge set

$$\begin{array}{rcl} E & \setminus & \{\ \{x,y\} \in E \ :\ x \in \{u,w\},\ y \in V\ \} \\ & \cup & \{\ \{v_{uw}, x\}\ :\ \{x,u\} \in E\ \vee\ \{x,w\} \in E\ \}. \end{array}$$

A graph H is a *contraction* of a graph G (or G is *contractible* to H) if H can be obtained from G by a sequence of edge contractions.

1.1 Previous work.

The problem of checking whether a graph is a contraction of another has already attracted some attention. In this subsection we briefly survey known results. For a fixed graph H, the algorithmic H-CONTRACTIBILITY problem is to decide if the input graph is contractible to H.

Stars and triangle-free patterns.

Perhaps the first systematic study of contractions was undertaken by Brouwer and Veldman [2]. Here are two main theorems from that paper.

Theorem 1 (Theorem 3 in [2])**.** *A graph G is contractible to $K_{1,m}$ if and only if G is connected and contains an independent set S of m vertices such that $G - S$ is connected.*

In particular, a graph is contractible to P_3 if and only if it is connected and is neither a cycle nor a complete graph. The theorem also allows to detect, in polynomial time, if a graph is contractible to $K_{1,m}$. It suffices to enumerate over all sets S with m independent vertices and check if the graph $G - S$ is connected. This gives an $|V(G)|^{\mathcal{O}(m)}$ algorithm for $K_{1,m}$-CONTRACTIBILITY (in particular for P_3-CONTRACTIBILITY), which is polynomial for every fixed m.

Theorem 2 (Theorem 9 in [2])**.** *If H is a connected triangle-free graph other than a star, then H-CONTRACTIBILITY is NP-complete.*

Hence, checking if a graph is contractible to P_4 or C_4 is NP-complete. More generally, it is NP-complete for every bipartite graph with at least one connected component that is not a star.

Patterns up to 5 vertices.

The research direction initiated by Brouwer and Veldman was continued by Levin, Paulusma, and Woeginger [12], [13]. Here is the main result established in these two papers.

Theorem 3 (Theorem 3 in [12]). *Let H be a connected graph on at most 5 vertices. If H has a dominating vertex, then H-CONTRACTIBILITY can be decided in polynomial time. If H does not have a dominating vertex, then H-CONTRACTIBILITY is NP-complete.*

However, the existence of a dominating vertex in the pattern H is not enough to ensure that contractibility to H can be decided in polynomial time. A pattern on 69 vertices for which contractibility to H is NP-complete was exhibited in [10].

When the pattern is part of input.

Looking at contractions to fixed pattern graphs (H-CONTRACTIBILITY) is justified by the following theorems proved by Matoušek and Thomas in [15].

Theorem 4 (Theorem 4.1 in [15]). *The problem of deciding, given two input graphs G and H, whether G is contractible to H is NP-complete even if we impose one of the following restrictions on G and H:*

(i) *H and G are trees of bounded diameter,*

(ii) *H and G are trees all whose vertices but one have degree at most 5.*

Theorem 5 (Theorem 4.3 in [15]). *For every fixed k, the problem of deciding, given two input graphs G and H, whether G is contractible to H is NP-complete even if we restrict G to partial k-trees and H to k-connected graphs.*

The authors also proved a positive result.

Theorem 6 (Theorem 5.14 in [15]). *For every fixed Δ, k, there exists an $\mathcal{O}(|V(H)|^{k+1} \cdot |V(G)|)$ algorithm to decide, given two input graphs G and H, whether G is contractible to H, when the maximum degree of H is at most Δ and G is a partial k-tree.*

Cyclicity.

The *cyclicity* of a graph G is defined as the largest integer k for which G is contractible to a cycle on k vertices. Exact values for some graphs and lower and upper bounds for some classes of graphs are given by Hammack in [9]. He also presented a polynomial-time algorithm for computing cyclicity of a planar graph.

Non-recursive classes closed under taking of contractions.

Another type of containment relation close to contractions – induced minors – was studied by Matoušek, Nešetřil, and Thomas in [14]. A graph is an *induced minor* of another if the first is a contraction of an induced subgraph of the latter. The authors of [14] prove (Theorem 1.8) that there exists a class closed under taking of induced minors which is non-recursive (i.e. there is no algorithm to test the membership in this class). Clearly, a class of graphs closed under taking of induced minors is also closed under taking of contractions (and induced subgraphs) so we restate their result in the following way.

Theorem 7 (Theorem 1.18 in [14]). *There exists a non-recursive class of graphs closed under taking of contractions (and induced subgraphs).*

Wagner's Conjecture for contractions.

The statement usually referred to as *Wagner's Conjecture* is that for any infinite sequence $G_0, G_1, \ldots$ of graphs, there is a pair i, j such that $i < j$ and G_i is a minor of G_j. The proof of Wagner's Conjecture is one of the highlights of the Graph Minors project [22].

A contraction version of Wagner's Conjecture was considered by Demaine, Hajiaghayi, and Kawarabayashi in [3]. They disproved this version showing the following.

Theorem 8 (Theorem 31 in [3]). *There is an infinite sequence $G_0, G_1, \ldots$ of connected graphs such that, for every pair i, j ($i \neq j$), G_i is not a contraction of G_j.*

However, the authors also proved that the conjecture holds when the graphs in the sequence are required to be trees, or triangulated planar graphs, or 2-connected embedded outerplanar graphs.

1.2 Our contribution.

A graph is a *minor* of another if the first is a contraction of a subgraph of the latter. Graph Minors is a celebrated project by Robertson and Seymour that is considered to be an important part of modern Graph Theory. One of the algorithmic consequences of Graph Minors is that, for every graph H, there exists a cubic-time algorithm deciding whether the input graph contains H as a minor [20]; and another, for every class of graphs closed under taking minors, there exists a cubic-time algorithm deciding whether the input graph belongs to this class [22].

While graph minors are well-studied both from combinatorial and algorithmic point of view, relatively little is known about graph contractions which are rather close to graph minors. Algorithmically, they are much less tractable compared to minors. As mentioned in the previous subsection, there are graphs for which it is NP-complete to decide if the input graph is contractible to them; and there are non-recursive classes of graphs, that are closed under taking of contractions, where there is no algorithm deciding whether an input graph belongs to this class.

In this work we show, for a large class of inputs – graphs embeddable on surfaces – how to decide in polynomial time if a fixed graph is a contraction of the input. The key idea is to introduce surface versions of contractions and topological minors for graphs on surfaces. The difference between surface containment relations and the usual definition of contraction and topological minor is that the first respect the embedding. We show that a graph H embedded on a surface Σ is a surface contraction of a graph G embedded on Σ if the dual of H is a surface topological minor of the dual of G.

To use this duality algorithmically, we need to show that surface topological minors can also be found in polynomial time. This is done by reducing the problem of finding a surface topological minor to solving an instance of the disjoint path problem that can be solved in cubic time due to the main algorithmic result of Graph Minors [21].

2 Definitions

Basics.

We consider both simple graphs and multigraphs. When there is no ambiguity, "a graph" means a simple graph or multigraph. When we want to emphasize that multiple edges are allowed we say “a multigraph” and “a simple graph” if they are not allowed. For a (multi)graph G, let $V(G)$ be its vertex set and $E(G)$ its edge (multi)set. We allow graphs and multigraphs to have loops. Note that a loop contributes 2 to the degree of a vertex. For notation not defined here, we refer the reader to the monograph [4].

The geometric dual of a graph G embedded on a surface Σ will be denoted by G^*. [16, p. 103] The dual of a graph may be a multigraph. Notice that there is a one-to-one correspondence between the edges of G and the edges of G^*. We keep the convention that e^* is the edge of G^* corresponding to edge e of G.

A graph H is a *subdivision* of a graph G, when H can be obtained from G by subdividing its edges (i.e., replacing edges by paths). A graph H is a *topological minor* of a graph G if H is a subdivision of a subgraph of G. Vertices of degree ≥ 3 in a subdivision are called *branch vertices*.

In this paper we consider the algorithmic problem of contracting an input graph G to a fixed graph H. Below we will assume that both H and G are connected. This can be done without loss of generality. If G and H are not connected, we consider contracting different connected components of G to different connected components of H. Since H is fixed, this will only contribute to a constant (in $|V(G)|$) factor in the computational complexity of the algorithm.

Embeddings.

In this work, we only need to distinguish between essentially different embeddings of a graph embedded on a surface. This motivates the following definition.

Two graphs G and H embedded on a surface Σ are *combinatorially equivalent* ($G \simeq H$) if there exists a homeomorphism of Σ (in which they are embedded) that transforms one into the other. The relation of being combinatorially equivalent is reflexive, symmetric and transitive, and thus an equivalence relation. Let $\mathcal{G}$ be the class of all graphs embeddable on Σ and isomorphic to a graph G and let us consider the quotient set $\mathcal{G}/\simeq$. The equivalence classes (i.e., the elements of the quotient set) can be thought of as *embeddings*. In fact, we will work with embeddings, but for simplicity, we will pick a representative (embedded) graph for each embedding.

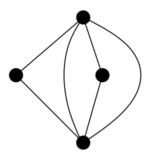

Figure 1: A thin triangulated planar graph.

Homotopic edges and thin graphs.

Two edges of an embedded graph are *homotopic edges* if they together bound a 2-face (a face incident with exactly 2 edges). Following [1], a *thin graph* is a multigraph embedded on a surface that has no homotopic pairs of edges (see Figure 1). It turns out that thin plane multigraphs cannot have more edges than simple maximal plane graphs.

Lemma 9 (Lemma 5 in [1])**.** *If G is a thin plane graph, then $|E(G)| \leq 3|V(G)| - 6$.*

We will prove that thin graphs embeddable on a surface Σ cannot have more than simple maximal graphs embeddable on Σ.

Lemma 10. *Let $g \geq 0$ be an integer and G be a graph with at least 3 edges embeddable on a surface of Euler genus g. If G is a thin graph, then $|E(G)| \leq 3(|V(G)| + 2g) - 6$.*

Proof. Let us remember the Euler's formula for graphs embedded on a surface. If $F(G)$ is the set of faces of G and g the Euler's genus of the surface, then $|F(G)| - |E(G)| + |V(G)| = 2(1 - g)$ [16, (4.1) & (4.4), p. 101].

Notice that if G is a maximal thin graph on a surface, then all its faces are bounded by three edges. Hence, $2 \cdot |E(G)| = 3 \cdot |F(G)|$. Substituting it back in the Euler's formula, we have that the number of edges of a maximal thin graph embeddable on a surface of Euler's genus g is,

$$|E(G)| = 3(|V(G)| + 2g) - 6.$$

The lemma follows. □

Surface containment relations.

A *surface contraction* of an edge e of an embedded graph G is an embedded graph G' that is obtained by homeomorphically mapping the endpoints of e

in G to a single vertex without any edge crossings and recursively removing one of two homotopic edges, if a graph has such a pair. Notice that there are many surface contractions of an edge of an embedded graph G but they are all combinatorially equivalent.

A *surface dissolution* of a vertex v of degree 2 in an embedded graph G is a surface contraction of one of the two edges v is incident with in G.

Let G and H be two embedded graphs. We say that H is a surface contraction of G ($H \leq_{ec} G$), if H is combinatorially equivalent to a graph that can be obtained from G by a series of surface contractions. We say that H is a *surface topological minor* of G ($H \leq_{etm} G$), if H is combinatorially equivalent to a graph that can be obtained from G by a series of vertex and edge deletions, and surface dissolution of vertices of degree 2.

3 Contractions vs topological minors

Lemma 11. *Let Σ be a surface. H and G be two thin graphs embedded on Σ and H^*, G^* their respective duals.*

$$H \leq_{ec} G \iff H^* \leq_{etm} G^*$$

Proof. Let G be a thin graph and e an edge of G. Let $G_{/e}$ be a surface contraction of e in G. Notice that ${G_{/e}}^*$ is isomorphic to an embedded graph obtained from G^* by deleting e^* and recursively applying surface dissolutions of vertices of degree 2. (Pairs of homotopic edges in an embedded graph correspond to vertices of degree 2 in its dual.) Let us also note that $G_{/e}$ is an embedded graph with no homotopic edges.

If H can be obtained from G by a series of surface contractions, then H^* can be obtained from G^* by a series of edge deletions and surface dissolutions of vertices of degree 2. Hence, if H is a surface contraction of G, then H^* is a surface topological minor of G^*. This proves the forward implication.

For the backward implication, suppose that H^* is a surface topological minor of G^*; that is H^* can be obtained from G^* by a sequence of vertex deletions, edge deletions, and surface dissolutions of vertices of degree 2.

Let us notice that removing a vertex v in a embedded graph can be simulated by removing all but two edges incident to v, then applying a surface dissolution to v and removing the new edge. Hence, a sequence of vertex deletions, edge deletions, and surface dissolutions of vertices of degree 2 can be replaced by a sequence of edge deletions and surface dissolutions of vertices of degree 2. The sequence can be rearranged and split into groups – every group consists of an edge removal and appropriate surface dissolutions of vertices of degree 2.

Each group of operations in an embedded graph corresponds to a surface edge contraction in its dual. The sequence of operations that transform G^* into H^* corresponds to a sequence of surface edge contractions that brings G into H. Hence, the backward implication. $\square$

A simple graph H embedded on a surface Σ is a *schema* of a multigraph H' embeddable on Σ, if $V(H) = V(H')$ and two vertices are adjacent in H if and only if they are adjacent in H'. In other words, a schema of the multigraph is the graph obtained by replacing multiple with single edges. Let $\mathcal{C}(H)$ be a maximal set of thin embedded (on Σ) multigraphs whose schema is H such that they are all combinatorially different.

Lemma 12. *Let Σ be a surface. For every graph H embeddable on Σ, the set $\mathcal{C}(H)$ is finite.*

Proof. Combinatorially different embeddings of a multigraph H embeddable on Σ are determined by cyclic orders of neighbors on vertices. There might be infinitely many embeddings of a multigraph embeddable on Σ. However, we are confined to thin embedded graphs only and each will have at most $3(|V(H)| + 2g) - 6$ edges by Lemma 10. Hence, the number of possible different cyclic orderings is finite. $\square$

Theorem 13. *Let Σ be a surface. Let H and G be simple graphs embeddable on Σ and $\widetilde{G}$ a graph embedded on Σ and isomorphic to G. Then,*

$$H <_c G \iff \exists\, \widetilde{H} \in \mathcal{C}(H) \text{ such that } \widetilde{H} <_{ec} \widetilde{G}.$$

Proof. For the backward implication, let H be the schema of some $\widetilde{H} \in \mathcal{C}(H)$. ($H$ is a simple graph.) Let us notice that if $\widetilde{H}$ is combinatorially equivalent to a surface contraction of $\widetilde{G}$, then $\widetilde{G}$ (and its abstract graph G) are isomorphic to a contraction of H.

For the forward implication, let us assume that $H <_c G$. There exists a sequence of edge contractions that brings G into H. Let us apply the same sequence as a sequence of surface contractions to $\widetilde{G}$ and call the resulting graph $\widetilde{T}$. From the definition of surface contraction, $\widetilde{T}$ is thin. Notice that its schema is H. From the choice of $\mathcal{C}(H)$, there exists $\widetilde{H} \in \mathcal{C}(H)$ that is combinatorially equivalent to $\widetilde{T}$. $\square$

A direct consequence of Lemma 11 and Theorem 13 is the following corollary.

Corollary 14. *Let Σ be a surface. Let H and G be graphs embeddable on Σ and $\widetilde{G}$ a graph embedded on Σ and isomorphic to G. Then,*

$$H <_c G \iff \exists\, \widetilde{H} \in \mathcal{C}(H) \text{ such that } \widetilde{H}^* <_{etm} \widetilde{G}^*.$$

4 Surface topological minors and the algorithm

In this section, we reduce the problem of finding a surface topological minor to the the problem of finding a collection of disjoint paths in a graph. Here is a result from Graph Minors we will need later.

Theorem 15 ([21])**.** *There exists an algorithm that given a graph G and k pairs $(s_1, t_1), \ldots, (s_k, t_k)$ of vertices of G decides whether there are k vertex-disjoint paths $P_1, \ldots, P_k$ in G such that P_i joins s_i and t_i, for all $i = 1, \ldots, k$, and if so, finds them. The algorithm runs in time $\mathcal{O}(|V(G)|^3)$.*

Theorem 16. *Let Σ be a surface. For every graph H embedded on Σ, there exists a polynomial-time algorithm that given a graph G embedded on Σ decides if H is a surface topological minor of G, and if so, finds the subgraph which is a subdivision of H.*

Proof. A k-star is a connected bipartite graph whose one part has one vertex (the *center*) and the other part has k vertices (the *leaves*). A star is a graph that is a k-star for some k. A *labelled star* is a subgraph of G that is a star and whose center is labelled with a vertex from $V(H)$ and whose leaves are labelled with different edges from $E(H)$ that are incident with v in H. A labelled star Q is said to be *compatible with* $v \in V(H)$ if it is a deg(v)-star, its center is labelled with v, and the cyclic ordering of the labels on the leaves of Q is the same as the cyclic ordering of the edges incident with v in H.

Let us fix an ordering $v_1, \ldots, v_{|V(H)|}$ of $V(H)$ and an ordering $e_1, \ldots, e_{|E(H)|}$ of $E(H)$. A *branching* is a $|V(H)|$-tuple $(Q_{v_i} : i = 1, \ldots, |V(H)|)$ such that Q_{v_i} for $i = 1, \ldots, |V(H)|$ is a labelled star compatible with v_i. A *good branching* is one in which no two centers of stars coincide. Let $\mathcal{Q}$ be the set of all different good branchings. Notice that $|\mathcal{Q}|$ is bounded by $|V(G)|^{\mathcal{O}(|V(H)|)}$.

For a branching from $\mathcal{Q}$ we define an instance of the disjoint path problem. We start with $|E(H)|$ pairs of terminals and later will be possibly removing some. For every $j = 1, \dots, |E(H)|$, let $\{s_j, t_j\}$ is the two vertices of G that are labelled with e_j in the branching. We then remove from the set of pairs such $\{s_j, t_j\}$ that s_j and t_j are adjacent. We then remove all centers of stars from G.

Claim. H is a surface topological minor of G if and only if there exists a branching from $\mathcal{Q}$ that defines a feasible disjoint path instance.

If H is a surface topological minor of G, then the branching is given by the set of stars centered at the branch vertices of H whose edges incident with the center are those that belong to the model of H in G.

If there is a branching in $\mathcal{Q}$ that defines a feasible disjoint path instance, then the union of the disjoint paths and the stars in the branching give a model of a surface topological minor of H in G.

Now we are ready to present an algorithm that for a fixed graph H decides if an embedded graph G contains H as a minor topological minor of G. First the algorithm constructs the set $\mathcal{Q}$. Then, for every branching from $\mathcal{Q}$ the algorithm constructs an instance of the disjoint paths problem and tests its feasibility.

The correctness of the algorithm is a direct consequence of the Claim. To see that the running time of the algorithm is polynomial in $|V(G)|$, notice that – as mentioned before – the cardinality of $\mathcal{Q}$ is bounded by $|V(G)|^{\mathcal{O}(|V(H)|)}$; building an instance of the disjoint paths problem out of a branching can be done in polynomial time; and testing feasibility of those instances can also be done in polynomial time by Theorem 15. □

Theorem 17. *Let Σ be a surface. For every graph H, there exists a polynomial-time algorithm that given a graph G embeddable on Σ decides whether H is a contraction of G, and if so finds a series of contractions transforming G into H.*

Proof. We can assume that both G and H are connected; otherwise, we consider contractions of different connected components of G to different connected components of H. We can also assume that H is embeddable on Σ since G can never be contracted to a graph that is not.

First we embed G on Σ using the linear-time algorithm from [17]. Let $\widetilde{G}$ be this embedded graph isomorphic to G and $\widetilde{G}^*$ its dual. For every graph

H from $\mathcal{C}(H)$, test if $\widetilde{H}^*$ is an embedded topological minor of $\widetilde{G}^*$, using the algorithm from Theorem 16.

The correctness of the algorithm follows from Corollary 14 and Lemma 12. The fact that the algorithm runs in polynomial time follows from Lemma 12 and Theorem 16. □

5 Discussion

We conclude with a number of remarks and a conjecture.

Solution via dual.

We prove our result by investigating what operation in the dual graph G^* corresponds to contractions in G. We want to mention that the same approach proved to be successful in studying maximum cuts in planar graphs. A maximum cut of a graph is its maximum bipartite subgraph. Orlova and Dorfman [19] and independently Hadlock [8] noticed that a (maximum) bipartite subgraph in G is an (maximum) Eulerian subgraph in G^*. Maximum Eulerian subgraphs can be found in polynomial time, therefore they proved that the maximum cut problem can be solved in polynomial-time in planar graphs.

Non-recursive classes of planar graphs.

We prove in this paper that for every graph H, there exists an algorithm that given a planar input graph G decides whether H is a contraction of G. To complement this result we would like to note that there are classes of planar graphs closed under taking of contractions that are non-recursive. A closer look at the proof of Theorem 7 in [14] reveals that the graphs in the non-recursive class from the theorem are in fact planar (and even have no K_5^- minor). We state it more formally.

Corollary 18. *There exists a non-recursive class of planar graphs closed under taking contractions.*

Cyclicity in bounded genus graphs.

Cycles have a unique embedding into the plane (up to combinatorial equivalence) and the dual of a cycle on k vertices is the multigraph with two vertices and k parallel edges. Therefore computing cyclicity of a plane graph is equivalent to solving the maximum flow problem in the dual for every pair of vertices (as the source and sink).

Hammack, also using the maximum flow problem, showed how to compute cyclicity of a planar graph [9]. Our result for bounded genus graphs allow to extend this result to graphs embeddable in an arbitrary suface.

Complexity of topological minor checking.

The running time of our algorithm checking if H is a surface topological minor of G is $\mathcal{O}(|V(G)|^{|V(H)|})$. From the point of view of parameterized complexity, that is, asking for an $f(|V(H)|) \cdot |V(G)|^{O(1)}$ step algorithm (i.e. classify it in the complexity class FPT when parameterized by the size of H), this is hardly satisfying. (We refer to [5, 18, 6] for more information on parameterized complexity.)

Recently, Grohe et al. announced an FPT algorithm for deciding if a fixed graph is a topological minor of an input graph [7]. However, their algorithm does not solve the problem we are interested in - surface topological minor. This is why we use the disjoint paths approach to obtain a polynomial-time (for every fixed H) algorithm.

Whether checking for surface topological minor belongs to the class FPT is not known even when the input graph is restricted to be embedded on a surface. We conjecture that this in fact is the case.

Conjecture. *For a surface Σ and a graph H, the problem of deciding whether H is a surface topological minor of a input graph G embedded on Σ is FPT, when parameterized by $|V(H)|$.*

6 Acknowledgements

We thank Samuel Fiorini for his kind support and Gwenaël Joret for stimulating discussions. We also gratefully acknowledge support from the Actions de Recherche Concertées (ARC) fund of the Communauté française de Belgique.

References

[1] Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004.

[2] A. E. Brouwer and H. J. Veldman. Contractibility and NP-completeness. *Journal of Graph Theory*, 11(1):71–79, 1987.

[3] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Improved grid minor bounds and Wagner's contraction. *Algorithmica*, 54(2):142–180, 2009.

[4] Reinhard Diestel. *Graph Theory*. Springer-Verlag, Electronic Edition, 2005.

[5] R.G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[6] J. Flum and M. Grohe. *Parameterized complexity theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.

[7] Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. `http://http://arxiv.org/abs/1011.1827`, 2010.

[8] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.*, 4(3):221–225, 1975.

[9] Richard Hammack. Cyclicity of graphs. *J. Graph Theory*, 32(2):160–170, 1999.

[10] Pim van 't Hof, Marcin Kamiński, Daniël Paulusma, Stefan Szeider, and Dimitrios M. Thilikos. On contracting graphs to fixed pattern graphs. In Jan van Leeuwen, Anca Muscholl, David Peleg, Jaroslav Pokorný, and Bernhard Rumpe, editors, *SOFSEM*, volume 5901 of *Lecture Notes in Computer Science*, pages 503–514. Springer, 2010.

[11] Marcin Kaminski, Daniël Paulusma, and Dimitrios M. Thilikos. Contractions of planar graphs in polynomial time. In Mark de Berg and Ulrich Meyer, editors, *ESA (1)*, volume 6346 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2010.

[12] Asaf Levin, Daniël Paulusma, and Gerhard J. Woeginger. The computational complexity of graph contractions I: Polynomially solvable and NP-complete cases. *Networks*, 51(3):178–189, 2008.

[13] Asaf Levin, Daniël Paulusma, and Gerhard J. Woeginger. The computational complexity of graph contractions II: Two tough polynomially solvable cases. *Networks*, 52(1):32–56, 2008.

[14] J. Matoušek, J. Nešetřil, and R. Thomas. On polynomial-time decidability of induced-minor-closed classes. *Comment. Math. Univ. Carolin.*, 29(4):703–710, 1988.

[15] Jirí Matousek and Robin Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. *Discrete Mathematics*, 108(1-3):343–364, 1992.

[16] B. Mohar and C. Thomassen. *Graphs on Surfaces.* The Johns Hopkins University Press, 2001.

[17] Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discrete Math.*, 12(1):6–26, 1999.

[18] Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications.* Oxford University Press, Oxford, 2006.

[19] G. Orlova and Y. Dorfman. Finding the maximum cut in a graph. *Tekhnicheskaya Kibernetika (Engineering Cybernetics)*, 10:502–506, 1972.

[20] Neil Robertson and Paul D. Seymour. Graph minors XII. Distance on a surface. *J. Comb. Theory, Ser. B*, 64(2):240–272, 1995.

[21] Neil Robertson and Paul D. Seymour. Graph minors XIII. The disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995.

[22] Neil Robertson and Paul D. Seymour. Graph minors XX. Wagner's conjecture. *J. Comb. Theory, Ser. B*, 92(2):325–357, 2004.

[23] Thomas Wolle and Hans L. Bodlaender. A note on edge contraction. Technical Report UU-CS-2004-028, Department of Information and Computing Sciences, Utrecht University, 2004.