Twin-Width of Graphs on Surfaces

Daniel Kráľ

Faculty of Informatics, Masaryk University, Brno, Czech Republic

Kristýna Pekárková

Faculty of Informatics, Masaryk University, Brno, Czech Republic

Kenny Štorgel

Faculty of Information Studies in Novo mesto, Slovenia

Abstract

Twin-width is a width parameter introduced by Bonnet, Kim, Thomassé and Watrigant [FOCS'20, JACM'22], which has many structural and algorithmic applications. Hliněný and Jedelský [ICALP'23] showed that every planar graph has twin-width at most 8. We prove that the twin-width of every graph embeddable in a surface of Euler genus g is at most $18\sqrt{47g}+O(1)$, which is asymptotically best possible as it asymptotically differs from the lower bound by a constant multiplicative factor. Our proof also yields a quadratic time algorithm to find a corresponding contraction sequence. To prove the upper bound on twin-width of graphs embeddable in surfaces, we provide a stronger version of the Product Structure Theorem for graphs of Euler genus g that asserts that every such graph is a subgraph of the strong product of a path and a graph with a tree-decomposition with all bags of size at most eight with a single exceptional bag of size $\max\{6, 32g-37\}$.

2012 ACM Subject Classification Mathematics of computing → Graphs and surfaces

Keywords and phrases twin-width, graphs on surfaces, fixed parameter tractability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.66

Related Version *Full Version*: `https://arxiv.org/abs/2307.05811`

Funding The first two authors have been supported by the MUNI Award in Science and Humanities (MUNI/I/1677/2018) of the Grant Agency of Masaryk University. The third author is supported by the Slovenian Research Agency (research program P1-0383, research projects J1-3002, J1-4008).

Acknowledgements The substantial part of the work presented in this article was done during the Brno–Koper research workshop on graph theory topics in computer science held in Kranjska Gora in April 2023, which all three authors have participated in. The authors would like to thank the anonymous reviewers for their numerous constructive comments, which helped to clarify and improve various aspects of the presentation.

1 Introduction

Twin-width is a graph parameter, which has recently been introduced by Bonnet, Kim, Thomassé and Watrigant [16, 17]. It has quickly become one of the most intensively studied graph width parameters due to its many connections to algorithmic and structural questions in both computer science and mathematics. In particular, classes of graphs with bounded twin-width (we refer to Section 2 for the definition of the parameter) include at the same time well-structured classes of sparse graphs and well-structured classes of dense graphs. Particular examples are classes of graphs with bounded tree-width, with bounded rank-width (or equivalently with bounded clique-width), and classes excluding a fixed graph as a minor. As the first order model checking is fixed parameter tractable for classes of graphs with bounded twin-width [16, 17], the notion led to a unified view of various earlier results on fixed parameter tractability of first order model checking of graph properties [21, 22, 30–33, 49], and more generally first order model checking properties of other combinatorial structures such as matrices, permutations and posets [4, 12, 19, 20].

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 66; pp. 66:1–66:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The foundation of the theory concerning twin-width has been laid by Bonnet, Kim, Thomassé and their collaborators in a series of papers [8–14,16,17], also see [51]. The amount of literature on twin-width is rapidly growing and includes exploring algorithmic aspects of twin-width [6,10,12,15,48], its combinatorial properties [2,4,7,10,14,24,47], and connections to logic and model theory [12,16,17,19,20,34]. While many important graph classes have bounded twin-width, good bounds are known only in a small number of specific cases. One of the examples is the class of graphs of bounded tree-width where an asymptotically optimal bound, exponential in tree-width, was proven by Jacob and Pilipczuk [40]. Another example is the class of planar graphs. The first explicit bound of 583 by Bonnet, Kwon and Wood [18] was gradually improved [5,35,40] culminating with a bound of 8 obtained by Hliněný and Jedelský [38]; also see [36,37] for a simpler proof and [41] for a promising approach of obtaining the upper bound of 7, which would be tight since Lamaison and the first author [42] constructed a planar graph with twin-width 7. In this paper, we extend this list by providing an asymptotically optimal upper bound on the twin-width of graphs embeddable in surfaces of higher genera. We prove the following two results (the latter is used to prove the former):

- We show that the twin-width of a graph embeddable in a surface of Euler genus g is at most $18\sqrt{47g}+O(1)$, which is asymptotically best possible; our proof also yields a quadratic time algorithm to find a witnessing sequence of vertex contractions.
- We provide a strengthening of the Product Structure Theorem for graphs embeddable in a surface of Euler genus g by showing that such graphs are subgraphs of a strong product of a path and a graph that almost has a bounded tree-width.

We next present the two results in more detail while also presenting the related existing results. While we prove both results in purely structural way, their proofs are algorithmic and yield a quadratic time algorithm (when the genus $g>0$ is fixed) that given a graph G embeddable in a surface of genus g, constructs a sequence of contractions witnessing that the twin-width of G is at most $18\sqrt{47g}+O(1)$. Further details are discussed in Section 5.

1.1 Twin-width of graphs embeddable in surfaces

Graphs that can be embedded in surfaces of higher genera, such as the projective plane, the torus and the Klein bottle, form important minor-closed classes of graphs with many applications and connections [45]. While the general theory concerning minor-closed classes of graphs yields that graphs embeddable in a fixed surface have bounded twin-width, the bounds are quite enormous: the results from [14, Section 4] on d-contractible graphs (graphs embeddable in a surface of Euler genus g are $O(g)$-contractible [39]) yields a bound double exponential in g, and the Product Structure Theorem for graphs embeddable in surfaces [26,28] together with results on the twin-width of graphs with bounded tree-width [40], of the strong product of graphs [46] and their subgraph closure [9] yields an exponential bound.

Bonnet, Kwon and Wood [18] showed that every graph embeddable in a surface of Euler genus g has twin-width at most $205g+583$. Our main result asserts that twin-width of every graph embeddable in a surface of Euler genus g is at most $18\sqrt{47g}+O(1)\approx 123.4\sqrt{g}+O(1)$. This bound is asymptotically optimal as any graph with $\sqrt{6g}-O(1)$ vertices can be embedded in a surface of Euler genus g and the n-vertex Erdős-Rényi random graph $G_{n,1/2}$ has twin-width at least $n/2-O(\sqrt{n\log n})$ [1], i.e., there exists a graph with twin-width $\sqrt{3g/2}-o(\sqrt{g})$ embeddable in a surface of Euler genus g. In particular, our upper bound asymptotically differs from the lower bound by a multiplicative factor $6\sqrt{282}\approx 100.76$. While several parts of our argument can be refined to decrease the multiplicative constant (to around 20), we have decided not to do so due to the technical nature of such refinements and the absence of additional structural insights gained by doing so.

1.2 Product Structure Theorem

On the way to our main result, we prove a modification of the Product Structure Theorem that applies to graphs embeddable in surfaces. The Product Structure Theorem is a recent significant structural result obtained by Dujmović, Joret, Micek, Morin, Ueckerdt and Wood [26, 28], which brought new substantial insights into the structure of planar graphs and led to breakthroughs on several long standing open problems concerning planar graphs, see, e.g. [25]. We also refer to the survey by Dvořák et al. [29] on the topic. The statement of the Product Structure Theorem originally proven by Dujmović et al. [26, 28] reads as follows (we remark that the statement in [26, 28] does not include the condition on planarity of the graph of bounded tree-width, however, an easy inspection of the proof yields this).

▶ **Theorem 1.** *Every planar graph is a subgraph of the strong product of a path and a planar graph with tree-width at most* 8*.*

Ueckerdt et al. [52] improved this as follows (we state a corollary of their main result to avoid defining the notion of simple tree-width, which is not needed in our further presentation).

▶ **Theorem 2.** *Every planar graph is a subgraph of the strong product of a path and a planar graph with tree-width at most* 6*.*

Dujmović et al. [26, 28] also proved two extensions of the Product Structure Theorem to graphs embeddable in surfaces.

▶ **Theorem 3.** *Every graph embeddable in a surface of Euler genus $g > 0$ is a subgraph of the strong product of a path, the complete graph K_{2g} and a planar graph with tree-width at most* 9*.*

▶ **Theorem 4.** *Every graph embeddable in a surface of Euler genus $g > 0$ is a subgraph of the strong product of a path, the complete graph $K_{\max\{2g,3\}}$ and a planar graph with tree-width at most* 4*.*

A stronger version was proven by Distel at el. [23]; the discussion of the even stronger statement implied by the proof of the next theorem given in [23] can be found after Theorem 6.

▶ **Theorem 5.** *Every graph embeddable in a surface of Euler genus $g > 0$ is a subgraph of the strong product of a path, the complete graph $K_{\max\{2g,3\}}$ and a planar graph with tree-width at most* 3*.*

We remark that it is not possible to replace K_{2g} in the statement of Theorems 3, 4 and 5 with a complete graph with $o(g)$ vertices as long as the bound on the tree-width stays constant since the layered tree-width of graphs embeddable in a surface of Euler genus g is linear in g [27] (the definition of layered tree-width is given in Section 2). To prove our upper bound on the twin-width of graphs embeddable in surfaces, we strengthen the statement of the Product Structure Theorem for graphs embeddable in surfaces as follows. Theorems 3, 4 and 5 imply that every graph embeddable in a surface of Euler genus $g > 0$ is a subgraph of the strong product of a path and a graph with tree-width at most $20g - 1$, $\max\{10g - 1, 14\}$ and $\max\{8g - 1, 11\}$, respectively. The next theorem, which we prove in Section 3, asserts that it is possible to assume that the tree-width of the graph in the product is almost at most 7 in the sense that all bags except possibly for a single bag have size at most 8.

▶ **Theorem 6.** *Every graph embeddable in a surface of Euler genus $g > 0$ is a subgraph of the strong product of a path and a graph H that has a rooted tree-decomposition such that*
- *the root bag has size at most $\max\{6, 32g - 37\}$, and*
- *every bag except the root bag has size at most* 8*.*

Similarly as it is not possible to replace K_{2g} with a complete graph with $o(g)$ vertices in Theorem 3, it is necessary to permit at least one of the bags to have a size linear in g in Theorem 6. Hence, the statement of Theorem 6 is the best possible asymptotically.

We remark that the proof of Theorem 5 given in [23] implies that every graph embeddable in a surface of Euler genus $g > 0$ is a subgraph of the strong product of a path and a graph that can be obtained from a planar graph with tree-width at most 3 by replacing one vertex of this planar graph with K_{2g} and the remaining vertices with K_3 (and replacing each edge of the planar graph with a complete bipartite graph between the corresponding sets of vertices). However, the vertex of the planar graph that is replaced with K_{2g} can be contained in many bags of the tree-decomposition and so the proof given in [23] does not yield a statement similar to that of Theorem 6 since the number of bags in the tree-decomposition with size linear in g can be arbitrary (although each such bag contains the same $2g$ vertices of K_{2g} in addition to 9 other vertices). The main new component in the proof of Theorem 6 (compared to the proofs given in [23, 26, 28]) is Lemma 12 given in Section 3, which is crucial so that we are able to restrict the sizes of all but one bag in a tree-decomposition to a constant size.

We also note the following corollary of Theorem 6 for projective planar graphs.

▶ **Corollary 7.** *Every graph embeddable in the projective plane is a subgraph of the strong product of a path and a graph with tree-width at most* 7.

2 Preliminaries

In this section, we introduce notation used throughout the paper. We use $[n]$ to denote the set of the first n positive integers, i.e., $\{1, \dots, n\}$. All graphs considered in this paper are simple and have no parallel edges unless stated otherwise; if G is a graph, we use $V(G)$ to denote the vertex set of G. A *triangulation* of the plane or a surface of Euler genus $g > 0$ is a graph embedded in such a surface such that every face is a 2-cell, i.e., homeomorphic to a disk, and bounded by a triangle. A *near-triangulation* is a 2-connected graph G embedded in the plane such that each inner face of G is bounded by a triangle.

We next give a formal definition of twin-width. A *trigraph* is a graph with some of its edges being red; the *red degree* of a vertex v is the number of red edges incident with v. If G is a trigraph and v and v' form a pair of its (not necessarily adjacent) vertices, then the trigraph obtained from G by *contracting* the vertices v and v' is the trigraph obtained from G by removing the vertices v and v' and introducing a new vertex w such that w is adjacent to every vertex u that is adjacent to at least one of the vertices v and v' in G and the edge wu is red if u is not adjacent to both v and v' or at least one of the edges vu and $v'u$ is red. The *twin-width* of a graph G is the smallest integer k such that there exists a sequence of contractions that reduces the graph G, i.e., the trigraph with the same vertices and edges as G and no red edges, to a single vertex, and none of the intermediate graphs contains a vertex of red degree more than k.

A *rooted tree-decomposition* $\mathcal{T}$ of a graph G is a rooted tree such that each vertex of $\mathcal{T}$ is a subset of $V(G)$, which we refer to as a *bag*, and that satisfies the following:

- for every vertex v of G, there exists a bag containing v,
- for every vertex v of G, the bags containing v form a connected subgraph (subtree) of $\mathcal{T}$, and
- for every edge e of G, there exists a bag containing both end vertices of e.

If the choice of the root is not important, we just speak about a *tree-decomposition* of a graph G. The *width* of a tree-decomposition $\mathcal{T}$ is the maximum size of a bag of $\mathcal{T}$ decreased by one, and the *tree-width* of a graph G is the minimum width of a tree-decomposition of G.

A *k-tree* is defined recursively as follows: the complete graph K_k is a k-tree and if G is a k-tree, then any graph obtained from G by introducing a new vertex and making it adjacent to any k vertices of G that form a complete subgraph in G is also a k-tree. Note that a graph G is a 1-tree if and only if G is a tree. More generally, a graph G has tree-width at most k if and only if G is a subgraph of a k-tree, and if G has at least k vertices, then G is a spanning subgraph of a k-tree. Note that k-trees have a tree-like structure given by their recursive definition, which also gives a rooted tree-decomposition of G with width k: the rooted tree-decomposition of K_k consists of a single bag containing all k vertices, and the rooted tree-decomposition of the graph obtained from a k-tree G by introducing a vertex w can be obtained from the rooted tree-decomposition $\mathcal{T}_G$ of G by introducing a new bag containing w and its k neighbors and making this bag adjacent to the bag of $\mathcal{T}_G$ that contains all k neighbors of w (such a bag exists since the subtrees of a tree have the Helly property).

A *BFS spanning tree* T of a (connected) graph G is a rooted spanning tree such that the path from the root to any vertex v in T is the shortest path from the root to v in G; in particular, a BFS spanning tree can be obtained by the breadth-first search (BFS). A *layering* is a partition of a vertex set of a graph G into sets $V_1, \ldots, V_k$, which are called *layers*, such that every edge of G connects two vertices of the same or adjacent layers. i.e., layers whose indices differ by one. If T is a BFS spanning tree of G, then the partition of $V(G)$ into sets based on the distance from the root of T is a layering. A *BFS spanning forest* F of a (not necessarily connected) graph G is a rooted spanning forest, i.e., a forest consisting of rooted trees, such that there exists a layering $V_1, \ldots, V_k$ compatible with F, i.e., for every tree of F, there exists d such that the vertices at distance ℓ from the root are contained in $V_{d+\ell}$. Note that if G is a graph and T a BFS spanning tree of G, then removing the same vertices in G and T results in a graph G' and a BFS spanning forest of G'. Finally, the *layered tree-width* of a graph G is the minimum k for which there exists a tree-decomposition $\mathcal{T}$ of G and a layering such that every bag of $\mathcal{T}$ contains at most k vertices from the same layer.

Consider a graph G and a BFS spanning tree T of G. A *vertical* path is a path contained in T with no two vertices from the same layer, i.e., a subpath of a path from a leaf to the root of T. The vertex of a vertical path closest to the root is its *top* vertex and the vertex farthest is its *bottom* vertex. Vertical paths with respect to a BFS spanning forest are defined analogously. If $\mathcal{P}$ is a partition of the vertex set of G to vertical paths, the graph $G/\mathcal{P}$ is the graph obtained by contracting each of the paths contained in $\mathcal{P}$ to a single vertex; note that the vertices of $G/\mathcal{P}$ can be viewed as the vertical paths contained in $\mathcal{P}$ and two vertical paths P and P' are adjacent in $G/\mathcal{P}$ if the graph G has an edge between $V(P)$ and $V(P')$.

3 Product Structure Theorem for graphs on surfaces

In this section, we provide the version of the Product Structure Theorem for graphs on surfaces, which we need to prove our upper bound on the twin-width. We start with recalling the following lemma proven by Dujmović et al. [26, 28]; note that the fundamental cycles determined by the edges $a_1b_1, \ldots, a_gb_g$ generate the fundamental group of the surface Σ.

▶ **Lemma 8.** *Let G be a triangulation of a surface Σ of Euler genus $g > 0$ and let T be a BFS spanning tree of G. There exist edges $a_1b_1, \ldots, a_gb_g$ not contained in the tree T with the following property. Let F_0 be the subset of edges of G comprised of the g edges $a_1b_1, \ldots, a_gb_g$ and the edges of the $2g$ paths in T from the root of T to the vertices $a_1, \ldots, a_g$ and $b_1, \ldots, b_g$. The surface Σ after the removal of the edges contained in F_0 is homeomorphic to a disk and its boundary is formed by a closed walk comprised of the edges contained in F_0.*

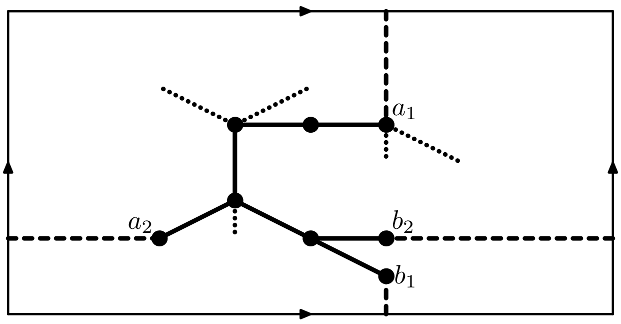

Figure 1 A rooted tree T_0 and edges a_1b_1 and a_2b_2, which are drawn dashed, bounding a part of the torus that is homeomorphic to a disk as in Lemma 9. Possible additional edges of the BFS spanning tree T are drawn dotted.

Using Lemma 8, one can prove the following; we refer to Figure 1 for the illustration of the notation in the case of the torus. The proof of the lemma is omitted due to space constraints.

▶ **Lemma 9.** *Let G be a triangulation of a surface of Euler genus $g > 0$ and let T be a BFS spanning tree of G. There exist a closed walk W in G, a subtree T_0 of T that contains the root of T, and k vertex-disjoint vertical paths $P_1, \dots, P_k$, $k \leq 2g$, such that*

- *the closed walk W bounds a part of the surface homeomorphic to a disk,*
- *the sets $V(P_1), \dots, V(P_k)$ form a partition of $V(T_0)$, i.e., $V(T_0) = V(P_1) \cup \cdots \cup V(P_k)$, and*
- *the sequence of vertices given by traversing the closed walk W can be split into at most $6g - 1$ segments such that all vertices of each segment belong to the same vertical path.*

Lemma 9 is one of two key ingredients for the proof of Theorem 13. The second, which is Lemma 12, relates to partitioning disk regions bounded by vertical paths. Similarly to [26, 28, 52], we make use of Sperner's Lemma, see e.g. [3, 50].

▶ **Lemma 10.** *Let G be a near-triangulation. Suppose that the vertices of G are colored with three colors in such a way that the vertices of each of the three colors on the outer face are consecutive, i.e., they form a non-empty path. There exists an inner face that contains one vertex of each of the three colors.*

The proof of the next lemma follows the lines of the proof of [52, Lemma 8] and is omitted due to space constraints. We say that a cycle is *covered* by paths $P_1, \dots, P_k$ if each path is a subpath of the cycle and each vertex of the cycle belongs to one of the paths $P_1, \dots, P_k$.

▶ **Lemma 11.** *Let G be a near-triangulation and let T be a BFS rooted spanning forest such that all roots of T are on the outer face. If the boundary cycle of the outer face can be covered by at most 6 vertex-disjoint vertical paths, say $P_1, \dots, P_k$, $k \leq 6$, then there exists a collection $\mathcal{P}$ of vertex-disjoint vertical paths such that*

- *the collection $\mathcal{P}$ contains the paths $P_1, \dots, P_k$,*
- *every vertex of G is contained in one of the paths in $\mathcal{P}$, and*
- *$G/\mathcal{P}$ has a rooted tree-decomposition of width at most seven such that the root bag contains the k vertices corresponding to the paths $P_1, \dots, P_k$.*

To state the next lemma, which is the second ingredient to prove the main result of this section, we need the following definition: if G' is a subgraph of a graph G embedded in a surface, a face of G' is a *region* if its interior contains a vertex or an edge of G; an *inner region* is an inner face that is a region. The proof of the lemma is omitted due to space constraints.

▶ **Lemma 12.** *Let G be a near-triangulation and let T be a BFS rooted spanning forest such that all roots of T are on the outer face. If the boundary cycle of the outer face can be covered by $k \geq 6$ vertex-disjoint vertical paths $P_1, \ldots, P_k$, then there exist a 2-connected subgraph G' of G and a collection $\mathcal{P}$ of vertex-disjoint vertical paths such that*

- *$\mathcal{P}$ contains the vertical paths $P_1, \ldots, P_k$,*
- *$\mathcal{P}$ contains at most $\max\{6, 6k - 32\}$ vertical paths,*
- *the vertex set of G' is the union of the vertex sets of the vertical paths contained in $\mathcal{P}$,*
- *the graph G' contains the boundary of the outer face,*
- *the graph G' has at most $\max\{1, 3k - 18\}$ inner regions, and*
- *the boundary cycle of each inner region of G' can be covered by at most six paths such that each is a subpath of a path from $\mathcal{P}$.*

We can now prove the main result of this section. Since a triangulation G in Theorem 13 is a subgraph of the strong product of a path and the graph $G/\mathcal{P}$, Theorem 13 readily implies Theorem 6. We remind that a tree-decomposition of $G/\mathcal{P}$ is a tree whose vertices are bags containing paths from the set $\mathcal{P}$ (the vertices of $G/\mathcal{P}$ can be viewed as these paths).

▶ **Theorem 13.** *Let G be a triangulation of a surface of Euler genus $g > 0$ and let T be a BFS spanning tree of G. There exists a collection $\mathcal{P}$ of vertical paths that partition the vertex set of G and the graph $G/\mathcal{P}$ has a rooted tree-decomposition such that*

- *the root bag has size at most $\max\{6, 32g - 37\}$,*
- *the root bag has at most $6 \cdot \max\{1, 18g - 21\}$ children, and*
- *every bag except the root bag has size at most 8.*

Moreover, every subtree $\mathcal{T}'$ of the tree-decomposition formed by a child of the root and all its descendants satisfies the following:

- *the bags of $\mathcal{T}'$ contain at most six paths that are contained in the root bag, and*
- *if $P_1, \ldots, P_k$ are all paths from $\mathcal{P}$ that are contained in the bags of $\mathcal{T}'$ but not in the root bag, the subgraph induced by $V(P_1) \cup \cdots \cup V(P_k)$ has a component joined by an edge to each of the paths that are contained both in the root bag and in $\mathcal{T}'$.*

Proof. Fix a triangulation G of a surface of Euler genus $g > 0$ and a BFS spanning tree T of G. We apply Lemma 9 to obtain a closed walk W, a subtree T_0 of T and k vertex-disjoint vertical paths $P_1, \ldots, P_k$, $k \leq 2g$, with the properties given in Lemma 9. Let $\ell \leq 6g - 1$ be the number of segments that cover the closed walk W as in the statement of the lemma.

We first deal with the general case $\ell \geq 7$ (note that if $\ell \geq 7$, then $g \geq 2$). We apply Lemma 12 to the near-triangulation obtained by cutting along the closed walk W, the BFS spanning forest obtained from T_0 by duplicating the vertices contained in W as needed, and the ℓ vertical paths that corresponds to the segments that cover the closed walk W. We obtain a collection $\mathcal{P}_0$ of vertex-disjoint vertical paths that contains at most $5\ell - 32$ additional vertical paths and a 2-connected subgraph G' such that the boundary of each inner region of G' can be covered by at most six paths contained in $\mathcal{P}_0$. In addition, the number of inner regions of G', further denoted by f, is at most $3\ell - 18$. Since $\ell \leq 6g - 1$, we obtain that $\mathcal{P}_0$ contains at most $30g - 37$ additional vertical paths and that $f \leq 18g - 21$. We now identify the duplicated vertices of T_0, i.e., G' has been modified to a subgraph of G, and we replace in the collection $\mathcal{P}_0$ the ℓ paths that cover the closed walk W with the paths $P_1, \ldots, P_k$. Hence, the size of the collection $\mathcal{P}_0$ is at most $32g - 37$ (note that $k \leq 2g$) and the boundary of each region of G' is still covered by at most six paths such that each is a subpath of the vertical paths contained in $\mathcal{P}_0$ (two different paths can be subpaths of the same vertical path).

If $\ell \leq 6$ (and so $k \leq 6$), we set $\mathcal{P}_0$ to be the collection $\{P_1, \ldots, P_k\}$ and G' the graph consisting of the vertices and the edges of the closed walk W; note that the only face of G' bounds a near-triangulation in G and $f = 1$.

We now proceed jointly for all values of ℓ. Suppose there is a region of G' such that the subgraph of G induced by the vertices of G inside this region does not have a component joined by an edge to each of the (at most six) paths that cover the boundary of the region and that are subpaths of paths from $\mathcal{P}_0$. Then, because G is a triangulation, there are two vertices on the boundary of this region joined by an edge not contained in G' and we add this edge to G'. We proceed as long as such a region exists and eventually obtain a graph G'' with $f' \leq 6f$ regions such that the boundary of each region can be covered by at most six paths, each subpath of a path contained in $\mathcal{P}_0$, and each region contains a component that is joined by an edge to each of the (at most six) paths that cover its boundary. We now apply Lemma 11 to each of the f' near-triangulations bounded by the regions of G'' and obtain rooted tree-decompositions $\mathcal{T}_1, \ldots, \mathcal{T}_{f'}$ with width at most seven of each them. Let $\mathcal{P}$ be the collection of vertical paths obtained from $\mathcal{P}_0$ by including all additional vertical paths obtained by these f' applications of Lemma 11.

We now construct a rooted tree-decomposition of $G/\mathcal{P}$. The root bag contains the vertices corresponding to the paths in $\mathcal{P}_0$ and the subtrees rooted at its children are $\mathcal{T}_1, \ldots, \mathcal{T}_{f'}$. So, the root bag has size $|\mathcal{P}_0| \leq \max\{6, 32g - 37\}$, it has $f' \leq 6f \leq 6\max\{1, 18g - 21\}$ children, and all bags except the root bag has size at most 8. Consider now a subtree $\mathcal{T}_i$, $i \in [f']$. The only paths from $\mathcal{P}_0$ contained in the bags of $\mathcal{T}_i$ are the at most six paths whose subpaths cover the boundary of the corresponding region of G'' and the vertices contained in the paths of the bags of $\mathcal{T}_i$ but not in the paths of $\mathcal{P}_0$ are exactly the vertices of G contained inside the region. Hence, the subgraph of G induced by such vertices has a component joined by an edge to each of the paths from $\mathcal{P}_0$ contained in the root bag of $\mathcal{T}_i$. We conclude that the obtained rooted tree-decomposition of $G/\mathcal{P}$ has the properties given in the statement. ◀

4 Bound on twin-width

We now present the asymptotically optimal upper bound on the twin-width of graphs embeddable in surfaces.

▶ **Theorem 14.** *The twin-width of every graph G of Euler genus $g \geq 1$ is at most*

$$6 \cdot \max\left\{3\sqrt{47g} + 1, 2^{24}\right\} = 18\sqrt{47g} + O(1).$$

Proof. Fix a graph G of Euler genus $g > 0$ and let G_0 be any triangulation of the surface with Euler genus g that G is a spanning subgraph of G_0, i.e., $V(G_0) = V(G)$ (to avoid unnecessary technical issues related to adding new vertices, G_0 may contain parallel edges).

We apply Theorem 13 to G_0 and an arbitrary BFS spanning tree T_0; let $\mathcal{P}$ be a collection of vertical paths and $\mathcal{T}$ a rooted tree-decomposition as in Theorem 13. Let $P_1, \ldots, P_k$ be the vertical paths contained in the root bag and let $\mathcal{T}_1, \ldots, \mathcal{T}_\ell$ be the subtrees rooted at the children of the root bag (note that $k \leq 32g$ and $\ell \leq 108g$). Further, let V_i, $i \in [\ell]$, be the vertices contained in the vertical paths in the bags of $\mathcal{T}_i$ that are not contained in the root bag. Note that for every $i = 1, \ldots, \ell$, the subgraph induced by a set V_i has a component that is joined by an edge to each path P_j, $j \in [k]$, that is contained in a bag of the subtree $\mathcal{T}_i$.

Let H_0 be the graph obtained from G_0 by contracting each of the $k+\ell$ sets $V(P_1),\dots,V(P_k)$ and $V_1,\dots,V_\ell$ to a single vertex. Let $a_1,\dots,a_k$ and $b_1,\dots,b_\ell$ be the resulting vertices. Observe that H_0 can be obtained from G_0 by contracting edges and deleting vertices. Indeed, the vertices $a_1,\dots,a_k$ are obtained by contracting paths $P_1,\dots,P_k$ and each vertex b_i, $i=1,\dots,\ell$, can be obtained as follows: first contract the component of the subgraph induced by the set V_i that is joined by an edge to each of the paths $P_1,\dots,P_k$ contained in the subtree $\mathcal{T}_i$ to a single vertex, and then delete all the vertices of V_i not contained in this component. Since H_0 can be obtained from G_0 by contracting edges and deleting vertices, the graph H_0 can be embedded in the same surface as G_0. Hence, the number of the edges of H_0 is at most $3(k+\ell)-6+3g\le 3(k+\ell+g)$; the latter bound applies even if $k+\ell=2$. Since each subtree $\mathcal{T}_i$ contains at most six of the paths $P_1,\dots,P_k$, each of the vertices $b_1,\dots,b_\ell$ has degree at most six and all its (at most six) neighbors are among the vertices $a_1,\dots,a_k$.

Let $s=3\sqrt{47g}$; note that $s\ge 6$. We next split the vertices $a_1,\dots,a_k$ into sets $A_1,\dots,A_{k'}$ and the vertices $b_1,\dots,b_\ell$ into sets $B_1,\dots,B_{\ell'}$ as follows; a similar argument has also been used in [2]. Keep adding the vertices $b_1,\dots,b_\ell$ to the set B_1 until the sum of their degrees just exceeds s, then keep adding the remaining vertices to the set B_2 until the sum of their degrees just exceeds s, etc. Observe that the sum of the degrees of the vertices in each of the sets $B_1,\dots,B_{\ell'}$ is at most $s+6\le 2s$ and the sum of the degrees of the vertices in each of the sets $B_1,\dots,B_{\ell'-1}$ is at least s. Each of the vertices $a_1,\dots,a_k$ with degree larger than s forms a set of size one, and the remaining vertices are split in the same way as the vertices $b_1,\dots,b_k$. Each of the sets $A_1,\dots,A_{k'}$ has either size one or the sum of the degrees of its vertices is at most $2s$, and the sum of the degrees of the vertices in each of the sets $A_1,\dots,A_{k'-1}$ is at least s. Let H_0' be the graph obtained from H_0 by contracting the vertices in each of the sets $A_1,\dots,A_{k'}$ and each of the sets $B_1,\dots,B_{\ell'}$ to a single vertex; note that the graph H_0' does not need to be embeddable in the same surface as H_0. Since the sum of the degrees of the vertices $a_1,\dots,a_k$ and $b_1,\dots,b_\ell$ is at most $6(k+\ell+g)\le 846g$ (as H_0 has at most $3(k+\ell+g)$ edges), we obtain that $k'+\ell'\le\frac{846g}{s}+2=2s+2$, i.e., H_0' has at most $2s+2$ vertices.

We now describe the order in which we contract the vertices of G, and we analyze the described order later. In what follows, when we say a *layer*, we always refer to the layers given by the BFS spanning tree T_0 from the application of Theorem 13. In particular, each vertex of G is adjacent only to the vertices in its own layer and the two neighboring layers. To make the presentation clearer, we split contracting vertices into three phases.

Phase I. This phase consists of ℓ subphases. In the i-th subphase, $i\in[\ell]$, we contract all the vertices of the set V_i that are contained in the same layer to a single vertex in the way that we now describe. Then, we possibly contract them to some of the vertices created in the preceding subphases, i.e., those obtained by contracting vertices in $V_1\cup\dots\cup V_{i-1}$. In this phase, *we never contract two vertices contained in different layers* and no contraction involves any vertex from $V(P_1)\cup\dots\cup V(P_k)$.

Subphase. Fix $i\in[\ell]$. Let G_i be the subgraph of $G_0/\mathcal{P}$ induced by the vertices contained in the bags of the subtree $\mathcal{T}_i$ and let n' be the number of the paths $P_1,\dots,P_k$ that are contained in the bags of the subtree $\mathcal{T}_i$; note that $n'\le 6$. If the graph G_i has less than 8 vertices, we proceed directly to the conclusion of the subphase, which is described below. If the graph G_i has at least 8 vertices, G_i is a subgraph of a spanning subgraph of a 7-tree G_i' such that the n' vertices corresponding to the paths from the set $\{P_1,\dots,P_k\}$ are contained in the initial complete graph of G_i'.

Fix any order $Q_1,\dots,Q_n$ of the vertical paths corresponding to the vertices of G_i' such that the neighbors of Q_j, $j\in[n]$, among $Q_1,\dots,Q_{j-1}$ form a complete graph of order at most 7 in G_i' and the n' paths from the set $\{P_1,\dots,P_k\}$ are the paths $Q_1,\dots,Q_{n'}$. Let C_j be the complete subgraph of G_i formed by Q_j and its (at most 7) neighbors among $Q_1, \ldots, Q_{j-1}$. Note that the neighbors in G of each vertex of a path Q_j, $j \in [n]$, are contained in at most seven of the paths $Q_1, \ldots, Q_{j-1}$, which are exactly the paths forming the complete graph C_j. We define the *j-shadow* of a vertex $v \in V_i$ to be the set of its neighbors contained in the paths $Q_1, \ldots, Q_{j-1}$. Since every vertex of Q_j has at most 21 neighbors on the paths $Q_1, \ldots, Q_{j-1}$ (as its neighbors must be in the same or adjacent layers), the j-shadow of a vertex contained in the path Q_j has at most 21 vertices.

We now use the tree-like structure of the 7-tree G'_i to define the order of contractions of the vertices contained in V_i; this part of our argument is analogous to that used in [40] in relation to twin-width of graphs with bounded tree-width. We proceed iteratively for $j = n-1, \ldots, n'$. Before we describe the order of contractions, we present the properties satifised at the end of the iterations. At the end of the iteration for $j = n-1, \ldots, n'+1$, all vertices of V_i that

- are contained in paths of the same component of $G'_i \setminus \{Q_1, \ldots, Q_{j-1}\}$,
- have the same j-shadow, and
- are in the same layer

will have been contracted to a single vertex. At the end of the iteration for $j = n'$, all vertices of V_i with the same $(n'+1)$-shadow that are contained in the same layer will have been contracted to a single vertex. In particular, at the end of the iteration for $j = n'$, all vertices of V_i contained in the same layer will have been contracted to at most $2^{3n'}$ vertices (a vertex can have at most three neighbors on each path Q_j, $j \in [n']$, which are the vertex on the same layer and the two vertices on the adjacent layers, and the $(n'+1)$-shadow is a subset of these $3n'$ vertices).

For $j \in \{n-1, \ldots, n'\}$, we now describe the order of contractions of the vertices in the iteration for j. Let m be the number of components of $G'_i \setminus \{Q_1, \ldots, Q_j\}$ that are included in the component of $G'_i \setminus \{Q_1, \ldots, Q_{j-1}\}$ that contains Q_j, and let $W_1, \ldots, W_m$ be the sets of vertices obtained by contracting (in the previous iterations) vertices on the paths of these m components; note that the vertices in $W_1, \ldots, W_m$ are obtained by contracting some vertices contained in $V(Q_{j+1}) \cup \cdots \cup V(Q_n) \subseteq V_i$. Observe that each of the sets $W_1, \ldots, W_m$ has at most 2^{21} vertices in each layer (as the $(j+1)$-shadow of vertices on the same layer are subsets of the same set of $3 \cdot 7 = 21$ vertices). We first contract each vertex of W_2 to the vertex W_1 with the same $(j+1)$-shadow on the same layer if such vertex exists. Next, we contract each vertices of W_3 to the vertex of $W_1 \cup W_2$ with the same $(j+1)$-shadow on the same layer if such vertex exists, etc. At the end of this process, all vertices of $W_1 \cup \cdots \cup W_m$ with the same $(j+1)$-shadow that are on the same layer have been contracted to a single vertex (note that there are at most 2^{24} such vertices in each layer as the $(j+1)$-shadows are subsets of the same set of $3 \cdot 8 = 24$ vertices). If $j > n'$, we contract all resulting vertices with the same j-shadow that are on the same layer to a single vertex, and subsequently, we contract the vertex contained on the path Q_j to the vertex with the same j-shadow on the same layer (if such vertex exists). The description of the iteration for j is now finished.

Conclusion of subphase. The i-th subphase concludes by contracting all the vertices of V_i in the same layer to a single vertex, and if the vertex b_i is not the vertex with the smallest index in the set $B_{i'}$ such that $b_i \in B_{i'}$, i.e., $b_{i-1} \in B_{i'}$, then we contract each resulting vertex w to the vertex obtained in the $(i'-1)$-th subphase that is in the same layer as w (if such vertex exists).

Phase II. The graph obtained after Phase I has at most $k + \ell'$ vertices in each layer: k correspond to the vertices $a_1, \ldots, a_k$ of the graph H_0, i.e., they are contained on the paths $P_1, \ldots, P_k$, and the remaining ℓ' to the sets $B_1, \ldots, B_{\ell'}$ (see Figure 2). For every $i = 1, \ldots, k'$,

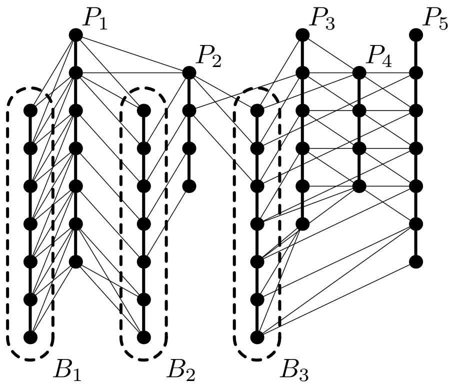

Figure 2 An example of a graph obtained after Phase I in the proof of Theorem 14 ($k = 5$ and $\ell' = 3$). The edges of vertical paths are drawn in bold. Note that there are no edges between paths corresponding to the set B_1, B_2 and B_3.

we contract all the vertices on the paths of A_i that are in the same layer to a single vertex as follows. Let $P_{i_1}, \ldots, P_{i_n}$ be the paths corresponding to the vertices of A_i. We first contract the vertices of P_{i_1} and P_{i_2} that are in the same layer, proceeding from top to bottom (starting with the layer that contains both such vertices). We then contract the vertices of P_{i_3} to the vertices created previously, again in each layer proceeding from top to bottom, then the vertices of P_{i_4}, etc. At the end of this phase, we obtain a graph that is a subgraph of the strong product of a path and the graph H_0'. Since H_0' has $k' + \ell' \le 2s + 2$ vertices, each layer now contains at most $2s + 2$ vertices.

Phase III. We now contract all the vertices contained in the top layer to a single vertex, then all the vertices of the next layer to a single vertex, etc. Finally, we contract the vertices one after another to eventually obtain a single vertex, starting with the two vertices of the top two layers, then contracting the vertex in the third layer, etc.

Analysis of red degrees. We now establish an upper bound on the maximum possible red degree of the vertices of the graphs obtained throughout the described sequence of contractions. We start with Phase I. During the i-th subphase and the iteration for j, the only new red edges ever created are among the vertices of $W_1, \ldots, W_m$ and the path Q_j. Since the vertices of $W_1 \cup \cdots \cup W_m$ have at most 2^{24} different $(j+1)$-shadows (the neighbors in their shadows are only on the paths contained in C_j), each vertex has neighbors in its and the two neighboring layers, and we first contract all vertices of $W_1 \cup W_2$ with the same $(j+1)$-shadow, then all vertices of $W_1 \cup W_2 \cup W_3$, etc., the red degree of any vertex does not exceed $2 \cdot 3 \cdot 2^{24} = 3 \cdot 2^{25}$. We eventually arrive at having at most 2^{24} vertices in each layer and so their red degrees do not exceed $3 \cdot 2^{24}$. Then, the vertices with the same j-shadow that are on the same layer are contracted, which can result in the vertices of Q_j (temporarily) having the red degree up to $3 \cdot 2^{24}$. At the end of iteration for $j > n'$, there are at most 2^{21} vertices in each layer that have been obtained from $W_1 \cup \cdots \cup W_m$ and so the red degree of each of them is at most $3 \cdot 2^{21}$. Also note that there is no red edge between the vertices on the paths $Q_1, \ldots, Q_{j-1}$ and the remaining vertices of V_i.

At the beginning of the conclusion of the subphase, each layer has at most 2^{18} vertices obtained from contracting the vertices of V_i (note that this bound also holds when G_i has less than eight vertices, i.e., when we proceeded directly to the conclusion of the subphase). The conclusion of the subphase starts with contracting these vertices to a single vertex per layer: this can increase the red degree of vertices on at most six paths $P_1, \ldots, P_k$ and the red degree of each vertex on these paths can increase by at most three. When the subphase finishes, each of the vertices contained in the paths $P_1, \ldots, P_k$ has at most $3\ell'$ red neighbors (although during the subphase it can have upto three additional red neighbors), and each of the vertices obtained by contracting the vertices of $V_1, \ldots, V_i$ has red degree at most $6s$ (since the sum of the degrees of the vertices in each set $B_1, \ldots, B_{\ell'}$ is at most $2s$). In particular, the red degree of each vertex on the paths $P_1, \ldots, P_k$ never exceeds $3(\ell' + 1)$. We conclude that the red degree of none of the vertices exceeds the largest of the following three bounds: $3 \cdot 2^{25}$, $3(\ell' + 1)$ and $6s$. Moreover, the red degree of no vertex exceeds $\max\{3\ell', 6s\} \leq 6s + 3$ (recall that $\ell' \leq 2s + 1$) at the end of each subphase (and so also at the end of Phase I).

During Phase II, each vertex has at most $\max\{k' + \ell', 2s\}$ red neighbors in its layer and in each of the neighboring layers. Indeed, the vertices obtained from those on the paths $P_1, \ldots, P_k$ have at most $k' + \ell'$ red neighbors in each layer (at most k' neighbors among vertices obtained from contracting vertices on the paths $P_1, \ldots, P_k$, and there are at most ℓ' vertices in each layer obtained by contracting vertices not on the paths $P_1, \ldots, P_k$) and the vertices obtained from those not on the paths $P_1, \ldots, P_k$ have at most $2s$ red neighbors in each layer as this is simply the upper bound on the number of their neighbors on the paths $P_1, \ldots, P_k$. Hence, during the entire Phase II, the red degree of any vertex never exceeds

$$3 \max\{k' + \ell', 2s\} \leq 3 \max\{2(s + 1), 2s\} = 6(s + 1).$$

Finally, since the number of vertices contained in each layer at the end of Phase II is at most $k' + \ell'$, during the entire Phase III, the red degree of no vertex exceeds $3(k' + \ell') - 1$.

Hence, we have established that the red degree of no vertex exceeds $\max\left\{6(s + 1), 3 \cdot 2^{25}\right\}$ during the whole process, which implies the bound claimed in the statement. ◀

5 Algorithmic aspects

We now overview the main steps of the algorithm based on the proof of Theorem 14 that computes a witnessing sequence of vertex contractions of a graph embeddable in a fixed surface. We remark that we measure the time complexity in terms of the number of vertices, and we recall the number of edges of an n-vertex graph embeddable in a surface of Euler genus g is at most $3n + 3g - 6$, i.e., linear in the number of vertices when g is fixed.

Since it is possible to find an embedding of a graph in a fixed surface in linear time [43, 44], we can assume that the input graph G is given together with its embedding in the surface. When the embedding of G in the surface is fixed, we complete it to a triangulation G' (we permit adding parallel edges if needed). We next choose an arbitrary BFS spanning tree T of G' and identify g edges $a_1b_1, \ldots, a_gb_g$ as described in Lemma 8, which was proven in [26, 28]. The proof of Lemma 8 in [26, 28] proceeds by constructing a spanning tree in the dual graph that avoids the edges of T and choosing the edges contained in neither T nor the spanning tree of the dual graph as the edges $a_1b_1, \ldots, a_gb_g$; this can be implemented in linear time. When the edges $a_1b_1, \ldots, a_gb_g$ are fixed, the construction of the walk W and the vertical paths described in Lemma 9 requires linear time.

We next compute the vertical paths described in Lemma 12 such that the boundary of each region of the graph G' obtained from the near-triangulation bounded by W can be covered by subpaths of at most six vertical paths. This requires processing the near-triangulation repeatedly following the steps of the inductive proof of Lemma 12: each step can be implemented in linear time and the number of steps is also at most linear. We then apply the recursive procedure described in the proof of Lemma 11 to each graph contained in one of the regions of G'; again, the number of steps in the recursive procedure is linear and each can be implemented in linear time. In this way, we obtain the collection $\mathcal{P}$ of vertical paths and the tree-decomposition $\mathcal{T}$ of $G'/\mathcal{P}$ described in Theorem 13. Note that the paths $\mathcal{P}$ and the tree-decomposition $\mathcal{T}$ fully determine the order of the contraction of the vertices and the order can be easily determined in linear time following the proof of Theorem 14.

We conclude that there is a quadratic time algorithm that constructs a sequence of contractions such that the red degree of trigraphs obtained during contractions does not exceed the bound given in Theorem 14. We remark that we have not attempted to optimize the running time of the algorithm, which would particularly require to implement the recursive steps of the proofs of Lemmas 11 and 12 more efficiently.

References

1 Jungho Ahn, Debsoumya Chakraborti, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Twin-width of random graphs, 2022. `arXiv:2212.07880`.

2 Jungho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Bounds for the twin-width of graphs. *SIAM Journal on Discrete Mathematics*, 36:2352–2366, 2022.

3 M. Aigner and G.M. Ziegler. *Proofs from THE BOOK*. Spinger, 2010.

4 Jakub Balabán and Petr Hliněný. Twin-width is linear in the poset width. In *16th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 214 of *LIPIcs*, pages 6:1–6:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

5 Michael A. Bekos, Giordano Da Lozzo, Petr Hliněný, and Michael Kaufmann. Graph product structure for h-framed graphs. In *33rd International Symposium on Algorithms and Computation (ISAAC)*, volume 248 of *LIPIcs*, pages 23:1–23:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

6 Pierre Bergé, Édouard Bonnet, and Hugues Déprés. Deciding twin-width at most 4 is NP-complete. In *49th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 229 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022.

7 Benjamin Bergougnoux, Jakub Gajarský, Grzegorz Guśpiel, Petr Hliněný, Filip Pokrývka, and Marek Sokołowski. Sparse graphs of twin-width 2 have bounded tree-width, 2023. `arXiv:2307.01732`.

8 Édouard Bonnet, Dibyayan Chakraborty, Eun Jung Kim, Noleen Köhler, Raul Lopes, and Stéphan Thomassé. Twin-width VIII: Delineation and win-wins. In *17th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 249 of *LIPIcs*, pages 9:1–9:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

9 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: Small classes. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1977–1996, 2021.

10 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width III: Max independent set, min dominating set, and coloring. In *48th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 198 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

11 Édouard Bonnet, Colin Geniet, Romain Tessera, and Stéphan Thomassé. Twin-width VII: Groups, 2022. `arXiv:2204.12330`.

12 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Toruńczyk. Twin-width IV: Ordered graphs and matrices. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*, pages 924–937, 2022.

13 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, and Stéphan Thomassé. Twin-Width V: Linear Minors, Modular Counting, and Matrix Multiplication. In *40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023)*, volume 254 of *LIPIcs*, pages 15:1–15:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

14 Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, and Stéphan Thomassé. Twin-width VI: The lens of contraction sequences. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1036–1056, 2022.

15 Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, Stéphan Thomassé, and Rémi Watrigant. Twin-width and polynomial kernels. In *16th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 214 of *LIPIcs*, pages 10:1–10:16. 2021.

16 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: Tractable FO model checking. In *61th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 601–612, 2020.

17 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: Tractable FO model checking. *Journal of the ACM*, 69:3:1–3:46, 2022.

18 Édouard Bonnet, O-joung Kwon, and David R. Wood. Reduced bandwidth: a qualitative strengthening of twin-width in minor-closed classes (and beyond), 2022. arXiv:2202.11858.

19 Édouard Bonnet, Jaroslav Nešetřil, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations, 2021. arXiv:2102.06880.

20 Édouard Bonnet, Jaroslav Nešetřil, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations. In *Proceedings of the 12th European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB'23)*, EUROCOMB, pages 156–162, Brno, Czech Republic, 2023. Masaryk University Press.

21 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

22 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.

23 Marc Distel, Robert Hickingbotham, Tony Huynh, and David R. Wood. Improved product structure for graphs on surfaces. *Discrete Mathematics & Theoretical Computer Science*, 24:8877:1–8877:10, 2022.

24 Jan Dreier, Jakub Gajarský, Yiting Jiang, Patrice Ossona de Mendez, and Jean-Florent Raymond. Twin-width and generalized coloring numbers. *Discrete Mathematics*, 345(3):112746, 2022.

25 Vida Dujmović, Louis Esperet, Gwenaël Joret, Bartosz Walczak, and David R. Wood. Planar graphs have bounded nonrepetitive chromatic number. *Advances in Combinatorics*, pages 5:1–5:11, 2020.

26 Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number. *Journal of the ACM*, 67(4):22:1–22:38, 2020.

27 Vida Dujmović, Pat Morin, and David R. Wood. Layered separators in minor-closed graph classes with applications. *Journal of Combinatorial Theory, Series B*, 127:111–147, 2017.

28 Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number. In *60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 862–875, 2019.

29 Zdeněk Dvořák, Tony Huynh, Gwenael Joret, Chun-Hung Liu, and David R. Wood. Notes on graph product structure theory. In *2019-20 MATRIX Annals*, volume 4 of *MATRIX Book Series*, pages 513–533. Springer, 2021.

30 Zdeněk Dvořák, Daniel Král', and Robin Thomas. Deciding first-order properties for sparse graphs. In *51th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 133–142, 2010.

31 Zdeněk Dvořák, Daniel Král', and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *Journal of the ACM*, 60(5):36:1–36:24, 2013.

32 Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposalbe graphs. In *26th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 1644 of *Lecture Notes in Computer Science*, pages 331–340. Springer, 1999.

33 Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM*, 48(6):1184–1206, 2001.

34 Jakub Gajarský, Michał Pilipczuk, and Szymon Toruńczyk. Stable graphs of bounded twin-width. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 39:1–39:12. ACM, 2022.

35 Petr Hliněný. Twin-width of planar graphs is at most 9, and at most 6 when bipartite planar, 2022. `arXiv:2205.05378`.

36 Petr Hliněný. Twin-width of planar graphs; a short proof, 2023. `arXiv:2302.08938`.

37 Petr Hliněný. Twin-width of planar graphs; a short proof. In *Proceedings of the 12th European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB'23)*, EUROCOMB, pages 595–600, Brno, Czech Republic, 2023. Masaryk University Press.

38 Petr Hliněný and Jan Jedelský. Twin-width of planar graphs is at most 8, and at most 6 when bipartite planar. In *50th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 261 of *LIPIcs*, pages 75:1–75:18. 2023.

39 J. Ivančo. The weight of a graph. *Annals of Discrete Mathematics*, 51:113–116, 1992.

40 Hugo Jacob and Marcin Pilipczuk. Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs. In *Graph-Theoretic Concepts in Computer Science (WG 2022)*, volume 13453 of *LNCS Lecture Notes Ser.*, pages 287–299. Springer International Publishing, 2022.

41 Jan Jedelský. *Twin-width of planar graphs.* Master's thesis, Masaryk University, 2024.

42 Daniel Kráľ' and Ander Lamaison. Planar graph with twin-width seven. *European Journal of Combinatorics*, pages 103749, 16pp, 2023.

43 Bojan Mohar. Embedding graphs in an arbitrary surface in linear time. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 392–397. ACM, 1996.

44 Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal on Discrete Mathematics*, 12(1):6–26, 1999.

45 Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces.* Johns Hopkins series in the mathematical sciences. Johns Hopkins University Press, 2001.

46 William Pettersson and John Sylvester. Bounds on the twin-width of product graphs. *Discrete Mathematics & Theoretical Computer Science*, 25:1, 2023.

47 Michał Pilipczuk and Marek Sokołowski. Graphs of bounded twin-width are quasi-polynomially χ-bounded. *Journal of Combinatorial Theory, Series B*, 161:382–406, 2023.

48 Michał Pilipczuk, Marek Sokołowski, and Anna Zych-Pawlewicz. Compact representation for matrices of bounded twin-width. In *39th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 219 of *LIPIcs*, pages 52:1–52:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022.

49 Detlef Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

50 E Sperner. Fifty years of further development of a combinatorial lemma. *Numerical solution of highly nonlinear problems*, pages 183–197, 1980.

51 Stéphan Thomassé. A brief tour in twin-width. In *49th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 229 of *LIPIcs*, pages 6:1–6:5. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022.

52 Torsten Ueckerdt, David R. Wood, and Wendy Yi. An improved planar graph product structure theorem. *Electronic Journal of Combinatorics*, 29(2), 2022.