Algorithmica (2019) 81:3655–3691
https://doi.org/10.1007/s00453-019-00592-7

Deleting Vertices to Graphs of Bounded Genus

Tomasz Kociumaka[1,2] · **Marcin Pilipczuk**[1]

Received: 13 June 2017 / Accepted: 28 May 2019 / Published online: 7 June 2019

Abstract
We show that a problem of deleting a minimum number of vertices from a graph to obtain a graph embeddable on a surface of a given Euler genus is solvable in time $2^{C_g \cdot k^2 \log k} n^{\mathcal{O}(1)}$, where k is the size of the deletion set, C_g is a constant depending on the Euler genus g of the target surface, and n is the size of the input graph. On the way to this result, we develop an algorithm solving the problem in question in time $2^{\mathcal{O}((t+g)\log(t+g))} n$ given a tree decomposition of the input graph of width t. The results generalize previous algorithms for the surface being a sphere by Marx and Schlotter (Algorithmica 62(3–4):807–822, 2012. https://doi.org/10.1007/s00453-010-9484-z), Kawarabayashi (in: 50th annual IEEE symposium on foundations of computer science, FOCS 2009, IEEE Computer Society, pp 639–648, 2009. https://doi.org/10.1109/FOCS.2009.45) and Jansen et al. (in: Chekuri (ed) 25th annual ACM-SIAM symposium on discrete algorithms, SODA 2014, SIAM, pp 1802–1811, 2014. https://doi.org/10.1137/1.9781611973402.130).

Keywords Fixed-parameter tractability · Bounded genus graphs · Bounded treewidth · Graph modification · Vertex deletion · Irrelevant vertex

1 Introduction

In recent years, a significant effort has been put into the study of parameterized complexity of recognizing near-planar graphs [7,8,12], that is, graphs that become planar after deleting a small number of vertices. Since, by the classic result of Lewis and

Research supported by Polish National Science Centre Grant DEC-2012/05/D/ST6/03214.

✉ Marcin Pilipczuk
malcin@mimuw.edu.pl

Tomasz Kociumaka
kociumaka@mimuw.edu.pl

[1] University of Warsaw, Warsaw, Poland

[2] Bar-Ilan University, Ramat Gan, Israel

Yannakakis [11], the decision version of the problem is NP-hard, it is natural to look for fixed-parameter algorithms with various parameters.

The parameter of the size of the deletion set naturally comes from the supposed applications: a number of efficient algorithms for planar graphs generalize well to near-planar graphs if one supplies them with the deletion set. Formally, we define the PLANAR VERTEX DELETION problem as follows: given a graph G and an integer k, decide if one can delete at most k vertices from G to obtain a planar graph.

Clearly, for a fixed integer k, the yes-instances to PLANAR VERTEX DELETION form a minor-closed graph class. Consequently, from the Graph Minors theory we obtain a nonuniform fixed-parameter algorithm for PLANAR VERTEX DELETION (cf. [2, Chapter 6]).

Marx and Schlotter [12] showed an explicit uniform fixed-parameter algorithm using a typical irrelevant vertex approach. First, they observe that the formulation of the problem as hitting all models of the forbidden minors for planar graphs (i.e., K_5 and $K_{3,3}$) leads to a fixed-parameter algorithm on bounded treewidth graphs by relatively standard techniques. Second, it is rather easy to believe (but quite technical to formally prove) that a middle part of a large, flat (planar), and grid-like subgraph of the input graph will always be disjoint with an optimal deletion set, and can be removed without changing the answer to the problem. The combination of the excluded grid theorem and the technique of iterative compression gives here a win-win approach: if the treewidth of the graph is not sufficiently bounded, an irrelevant part can be uncovered and removed.

There are two sources of potential inefficiencies in the approach of Marx and Schlotter. First, the routine for graphs of bounded treewidth that finds a minimum set hitting all forbidden minor models works in time double-exponential in the treewidth bound. Since the treewidth bound needs to be significantly larger than the size of the deletion set for the irrelevant vertex argument to work, we obtain at least a double-exponential dependency on the parameter. Second, the technique of iterative compression, at least applied in a straightforward manner, gives at least quadratic dependency on the input size.

Later, Kawarabayashi [8] designed a fixed-parameter algorithm with linear dependency on the input size. Finally, Jansen et al. [7] showed an algorithm with running time $2^{\mathcal{O}(k \log k)}n$, that is, with nearly single-exponential dependency on the parameter and linear dependency on the input size.

On high level, the work [7] follows the approach of Marx and Schlotter, but it improves upon both components. First, the authors describe a routine that explicitly constructs partial embeddings of graphs of bounded treewidth and solves the problem in question in time $2^{\mathcal{O}(t \log t)}n$ given a tree decomposition of width t. Second, they present arguments in the spirit of the aforementioned irrelevant vertex rule that reduce the graph to treewidth linearly bounded in the size of the solution (deletion set). Third, they circumvent the application of iterative compression by borrowing a trick from Bodlaender's algorithm for treewidth [1], where in a single step one compresses the graph by a multiplicative factor, yielding a linear dependency on the input size.

A simple reduction from the VERTEX COVER problem (replace every edge uv with a K_5 with vertices u, v, and 3 new vertices) shows that, unless the Exponential Time Hypothesis (ETH) [6] fails, the dependency on the parameter k needs to be

$2^{\Omega(k)}$ for any parameterized algorithm for PLANAR VERTEX DELETION. Although it is open whether PLANAR VERTEX DELETION can be solved in $2^{o(k \log k)} n^{\mathcal{O}(1)}$ time, we note that a lower bound by the second author [15] asserts that, unless the ETH fails, the bounded treewidth subroutine requires dependency $2^{\Omega(t \log t)}$ on the treewidth of the graph. This implies that a hypothetical algorithm that solves PLANAR VERTEX DELETION in $2^{o(k \log k)} n^{\mathcal{O}(1)}$ time needs to follow a significantly different approach compared to the one we know currently.

In the light of the aforementioned developments, in this paper, we initiate the study of the GENUS VERTEX DELETION problem: given a graph G and non-negative integers g and k, decide if one can delete at most k vertices from G to obtain a graph embeddable on a surface of Euler genus at most g.

Our main result is the following:

Theorem 1.1 *The* GENUS VERTEX DELETION *problem can be solved in time* $2^{C_g k^2 \log k} n^{\mathcal{O}(1)}$, *where* C_g *is a constant depending on* g *only.*

In the proof of Theorem 1.1, we follow the general approach of Marx and Schlotter [12]. Our main technical contribution is a generalization of the bounded treewidth subroutine of [7] to the bounded genus case.

Theorem 1.2 *Given a* GENUS VERTEX DELETION *instance* (G, g, k) *with* $|V(G)| = n$, *and a tree decomposition of* G *of width* t, *one can solve the* GENUS VERTEX DELETION *problem on* (G, g, k) *in time* $2^{\mathcal{O}((t+g)\log(t+g))} n$.

The proof of Theorem 1.2 follows the same principle as the corresponding routine of [7]—it builds partial embeddings for graphs with small boundaries—but its rigorous presentation requires overcoming significant technical hurdles.

The algorithm of Theorem 1.2 can be also seen as a generalization of an algorithm that computes the Euler genus in graphs of bounded treewidth: by setting $k = 0$, we obtain an algorithm with running time $2^{\mathcal{O}((t+g)\log(t+g))} n$ that checks if the input graph is embeddable on a surface of Euler genus at most g. Such a result is not new: a bounded treewidth routine is also a part of the current algorithms that compute embeddings in linear time [9,13]. In particular, the work of Kawarabayashi et al. [9] claims an algorithm with running time $f(t) \cdot n$ for some function f (i.e., with no dependency on g).[1]

For the second part, namely the irrelevant vertex argument, we generalize the approach of Marx and Schlotter [12]. In what follows, a set $S \subseteq V(G)$ is a *solution* to a GENUS VERTEX DELETION instance (G, g, k) if $|S| \leq k$ and $G - S$ is embeddable in a surface of Euler genus at most g.

Theorem 1.3 *There exists a sequence* $(C_g)_{g \geq 0}$ *of positive integers and an algorithm that, given a graph* G, *non-negative integers* g *and* k, *and a set* $M \subseteq V(G)$ *such that* $G - M$ *is embeddable into a surface of Euler genus at most* g, *in time* $C_g n^{\mathcal{O}(1)}$ *finds one of the following:*

[1] The work [9] is an extended abstract from FOCS 2008 that, to the best of our knowledge, never substantiated in a full version, and we were unable to reproduce the details of this algorithm from the description in [9].

1. *a tree decomposition of G of width at most $C_g|M|^{1/2}k^{3/2}$; or*
2. *a vertex $w \in V(G)$ such that every solution to the* Genus Vertex Deletion *instance (G, g, k) contains w; or*
3. *a vertex $v \in V(G)$ such that for every $S \subseteq V(G)\setminus\{v\}$ the set S is a solution to the* Genus Vertex Deletion *instance (G, g, k) if and only if S is a solution to a* Genus Vertex Deletion *instance $(G - \{v\}, g, k)$.*

Theorem 1.1 follows now from Theorems 1.2 and 1.3 in a usual manner. By a standard application of the iterative compression technique (cf. [2, Chapter 4]), we can assume that, apart from the input Genus Vertex Deletion instance (G, g, k), we are additionally given a set $M \subseteq V(G)$ of size $k+1$ such that $G-M$ is embeddable on a surface of Euler genus at most g (i.e., M is a slightly too large solution), at the cost of an additional $\mathcal{O}(n)$ factor in the running time bound. We iteratively apply the algorithm of Theorem 1.3 to (G, g, k) and M: if a vertex w or v is returned, we delete it and restart (decreasing the parameter k by one in case of a vertex w); if a tree decomposition is returned, we solve the Genus Vertex Deletion problem with the algorithm of Theorem 1.2.

Since $|M| = k+1$, the algorithm of Theorem 1.2 is applied to a tree decomposition of width of the order of $C_g k^2$, yielding the bound promised in Theorem 1.1.

While the lower bound of [15] shows that the bounded-treewidth routine of Theorem 1.2 has optimal running time (assuming ETH), we conjecture that the running time bound of Theorem 1.1 is not optimal, and can be improved similarly as it was in the planar case [7]. For this reason, we do not optimize the parameter dependency in Theorem 1.3, favouring the clarity of the arguments. In other words, we view our contribution as Theorem 1.2 being the main technical merit, with Theorem 1.3 and the resulting Theorem 1.1 being an example application.

The paper is organized as follows: after introducing notation for permutations and tree decompositions in Sect. 2, we discuss combinatorial embeddings in Sect. 3. Theorem 1.2 is proved in Sect. 4, and Theorem 1.3 is proved in Sect. 5.

2 Preliminaries

2.1 Permutations, Involutions, Cycles

For a non-negative integer t, we denote $[t] = \{1, 2, \ldots, t\}$. The group of all permutations of a set U is denoted by $\mathsf{Sym}(U)$. Given a set of permutations $S \subseteq \mathsf{Sym}(U)$, by $\langle S\rangle$ we denote the subgroup of $\mathsf{Sym}(U)$ generated by S. Given a subgroup Γ of the group of $\mathsf{Sym}(U)$, by $\mathsf{orb}(\Gamma)$ we denote the family of orbits of Γ. For a permutation σ, $\mathsf{orb}(\sigma)$ is a shorthand for $\mathsf{orb}(\langle\sigma\rangle)$; this also allows us to speak about orbits of a single permutation. An orbit is *trivial* if it consists of a single element.

A permutation $\sigma \in \mathsf{Sym}(U)$ is an *involution* if $\sigma(\sigma(i)) = i$ for every $i \in U$ and is *fixed-point free* if $\sigma(i) \neq i$ for every $i \in U$. Note that a permutation is a fixed-point free involution if and only if all its orbits are of size exactly two.

A permutation σ is a *cycle permutation* if it has exactly one nontrivial orbit; note that σ needs to act cyclically on this orbit. A *cycle* is an unordered pair consisting of a cycle permutation and its inverse.

We will need the operation of *restricting* a cycle permutation σ to a subset A of the elements of the non-trivial orbit v of σ: the result is a permutation σ_A, where $\sigma_A(e) = e$ for every $e \notin A$ while for every $e \in A$ we have $\sigma_A(e) = \sigma^k(e)$, where k is the minimum positive integer with $\sigma^k(e) \in A$. In other words, we shorten the nontrivial orbit v by crossing out the elements not belonging to A. Note that if $|A| \geq 2$, then σ_A is also a cycle permutation, while for $|A| \leq 1$ we have σ_A being an identity. The definition of restricting naturally extends to cycles by restricting both components. For a sequence P of elements of some set, by $\bar{P}$ we denote its reverse.

In our work, we will often analyze the subgroup of the permutation group spanned by two fixed-point free involutions. Observe that such an involution can be interpreted as a perfect matching of its domain (with edges representing orbits). Consequently, if $\alpha, \beta \in \mathsf{Sym}(U)$ are two fixed-point free involutions, then every orbit v of $\langle \alpha, \beta \rangle$ can be seen as a connected component of a union of two matchings, which is a cycle with alternating edges corresponding to α and β, respectively. In particular, $|v| = 2m$ is even and $(\beta \circ \alpha)^m(e) = e$ for every $e \in v$. Moreover, the orbit v consists of elements

$$e, \alpha(e), (\beta \circ \alpha)(e), \alpha \circ (\beta \circ \alpha)(e), \ldots, (\beta \circ \alpha)^{m-1}(e), \alpha \circ (\beta \circ \alpha)^{m-1}(e).$$

To fix notation, let us define a cycle permutation $o_e^{\langle \alpha, \beta \rangle}$ of U as follows:

$$o_e^{\langle \alpha, \beta \rangle}(f) = \begin{cases} \alpha(f) & \text{if } f = (\beta \circ \alpha)^i(e) \text{ for some } i, \\ \beta(f) & \text{if } f = \alpha \circ (\beta \circ \alpha)^i(e) \text{ for some } i, \\ f & \text{otherwise.} \end{cases}$$

Note that $o_e^{\langle \alpha, \beta \rangle}$ is a cycle permutation whose nontrivial orbit is v. Furthermore, while its definition formally depends on the choice of e and the order of α and β, different choices of $e \in v$ and a potential swap of the roles of α and β lead either to $o_e^{\langle \alpha, \beta \rangle}$ or its inverse. The definition of a cycle is suited to accommodate that: For the cycle permutation $o_e^{\langle \alpha, \beta \rangle}$, its cycle is denoted as $\hat{o}_e^{\langle \alpha, \beta \rangle}$. By the previous argumentation, the cycle does not depend on the order of α and β, nor on the choice of the element e within the same orbit.

Let $\hat{\sigma}_1$ and $\hat{\sigma}_2$ be two cycles of disjoint sets U_1 and U_2, with nontrivial orbits v_1 and v_2. A *merge* of $\hat{\sigma}_1$ and $\hat{\sigma}_2$ is a cycle $\hat{\sigma}$ of $U := U_1 \cup U_2$, whose nontrivial orbit v consists of exactly elements of $v_1 \cup v_2$. Furthermore, for some choice of cycle permutations $\sigma_1 \in \hat{\sigma}_1, \sigma_2 \in \hat{\sigma}_2, \sigma \in \hat{\sigma}$, for every $i = 1, 2$ and $e \in v_i$, if k is the minimum positive integer for which $\sigma^k(e) \in v_i$, then $\sigma^k(e) = \sigma_i(e)$. In other words, if we restrict the cycle of the nontrivial orbit of $\hat{\sigma}$ to the elements of U_i only, we obtain the cyclic order of the nontrivial orbit of $\hat{\sigma}_i$.

2.2 Tree Decompositions

A *tree decomposition* of a graph G is a pair $(\mathbf{T}, \beta)$, where $\mathbf{T}$ is a rooted tree and β is a function $\beta : V(\mathbf{T}) \to 2^{V(G)}$ that assigns to every $\mathbf{t} \in V(\mathbf{T})$ a *bag* $\beta(\mathbf{t}) \subseteq V(G)$ such that the following holds:

- for every edge $e \in E(G)$, there is a node $\mathbf{t} \in V(\mathbf{T})$ with $e \subseteq \beta(\mathbf{t})$;
- for every vertex $v \in V(G)$, the set $\{\mathbf{t} \in V(\mathbf{T}) : v \in \beta(\mathbf{t})\}$ is nonempty and induces a connected subgraph of $\mathbf{T}$.

The width of a decomposition is the maximum size of a bag, minus one.

For a tree decomposition $(\mathbf{T}, \beta)$ of a graph G, we define two auxiliary functions α and $G^{\downarrow}$ defined on the set of nodes $V(\mathbf{T})$. For a node $\mathbf{t} \in V(\mathbf{T})$,

- by $\alpha(\mathbf{t}) \subseteq V(G)$ we denote the union of all bags of the descendants of $\mathbf{t}$ (including $\mathbf{t}$ itself);
- by $G^{\downarrow}(\mathbf{t})$ we denote the graph $G[\alpha(\mathbf{t})] - E(G[\beta(\mathbf{t})])$, that is, the graph induced by $\alpha(\mathbf{t})$ with the edges inside the bag $\beta(\mathbf{t})$ removed.

For dynamic programming algorithms, it is often convenient to work with so-called *nice* tree decompositions, where the bag of the root is empty, and every node $\mathbf{t} \in V(\mathbf{T})$ is of one of the following four types:

leaf node has no children and its bag is empty;
introduce node has one child $\mathbf{t}'$ such that $\beta(\mathbf{t}) = \beta(\mathbf{t}') \cup \{v\}$ for some vertex $v \notin \beta(\mathbf{t}')$;
forget node has one child $\mathbf{t}'$ such that $\beta(\mathbf{t}) = \beta(\mathbf{t}') \setminus \{v\}$ for some vertex $v \in \beta(\mathbf{t}')$;
join node has two children $\mathbf{t}_1$ and $\mathbf{t}_2$ with $\beta(\mathbf{t}) = \beta(\mathbf{t}_1) = \beta(\mathbf{t}_2)$.

It is well known (see, e.g., [2,10]) that, in polynomial time, one can turn any tree decomposition of width t into an equivalent nice one having width at most t and $\mathcal{O}(n)$ bags in total.

2.3 Surfaces and Embeddings

The proof of Theorem 1.2 operates on an abstract notion of a hypergraph embedding defined and discussed in the next section. In the proof of Theorem 1.3, we rely on standard notions of a surface and an embedding as defined in the book of Mohar and Thomassen [14]. Since we do not use any nonstandard aspects of these notions, and since they are very intuitive, we refrain ourselves from redefining them here. We refer interested readers to [14, Chapters 3 and 4] for formal definitions of surfaces, embeddings, and genera. We remark that in Sect. 3 we discuss the relation between the abstract hypergraph embeddings used there and the (standard) graph embeddings used in the proof of Theorem 1.3.

3 Embeddings and Operations on them

3.1 Graph and Hypergraph Embeddings

We start with a clean but abstract notion of a hypergraph embedding, and then we restrict ourselves only to graph embeddings.

Definition 3.1 (*Hypergraph embedding*) A *hypergraph embedding* is a tuple $(\mathbf{F}, \theta, \sigma, \phi)$, where $\mathbf{F}$ is a finite set, whose elements are called *flags*, and θ, σ, and ϕ are three fixed-point free involutions of the set $\mathbf{F}$.

Given a hypergraph embedding $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$, we use the notation $\mathbf{F}(\mathcal{E}) := \mathbf{F}$ etc. Note that the number of flags needs to be even in any hypergraph embedding since θ, σ, and ϕ are fixed-point free involutions.

Definition 3.2 (*Connected components, vertices, edges, faces*) Let $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ be a hypergraph embedding. Then

a connected component is an orbit of $\mathsf{cc}_{\mathcal{E}} := \langle \theta, \sigma, \phi \rangle$;
a vertex is an orbit of $\mathtt{Verts}_{\mathcal{E}} := \langle \sigma, \phi \rangle$;
an edge is an orbit of $\mathtt{Edges}_{\mathcal{E}} := \langle \theta, \sigma \rangle$;
a face is an orbit of $\mathtt{Faces}_{\mathcal{E}} := \langle \theta, \phi \rangle$.

Note that, as all three permutations of $\mathcal{E}$ are fixed-point free involutions, every *object* (vertex, edge, or face) of $\mathcal{E}$ can be identified with a cycle, whose nontrivial orbit is the object in question.

Given a hypergraph embedding $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$, a *size-k* object is an orbit of $\mathtt{Edges}_{\mathcal{E}}$, $\mathtt{Verts}_{\mathcal{E}}$, or $\mathtt{Faces}_{\mathcal{E}}$ that consists of k flags. Note that k is always a positive even integer in this context. Also, observe that a size-2 edge corresponds to two equal orbits of θ and σ, a size-2 vertex corresponds to two equal orbits of σ and ϕ, while a size-2 face corresponds to two equal orbits of θ and ϕ.

We now define the genus of an embedding.

Definition 3.3 (*Genus of a hypergraph embedding*) Given a hypergraph embedding $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$, its *genus* is defined as

$$\hat{g}(\mathcal{E}) := \tfrac{1}{2}|\mathbf{F}| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| + 2|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})|. \tag{1}$$

Since the number of flags is even, the genus of a hypergraph embedding is always an integer.

Definition 3.4 (*Graph embedding*) A hypergraph embedding $\mathcal{E}$ is a *graph embedding* if every edge consists of exactly four flags.

In other words, a hypergraph embedding $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ is a graph embedding if no two orbits of θ and σ coincide, but the involutions θ and σ commute.

Observe that the formula for genus simplifies in case of a graph embedding.

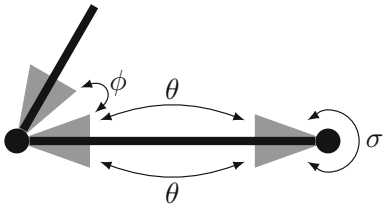

Fig. 1 Flags (gray triangles) and involutions in an embedding

Observation 3.5 If $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ is an embedding, then its genus equals

$$\hat{g}(\mathcal{E}) := |\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| + 2|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})|. \quad (2)$$

From this point, we use only graph embeddings in this work and call them simply *embeddings*.

If two objects share a flag, we say that these objects are *incident*. Note that a size-k object can be incident to at most $k/2$ objects of either of the other types (e.g., a size-k vertex can be incident to at most $k/2$ edges and $k/2$ faces). In particular, in an embedding, each edge can be incident to one or two faces and one or two vertices. An edge incident to only one vertex is called a *loop*.

Let us now relate the aforementioned definition of a (combinatorial) embedding with the topological intuition. Let G be a graph embedded on a surface. We visualize every edge as a (thin, and possibly bent) rectangle, with a flag attached at every corner of the rectangle (see Fig. 1). The involution θ pairs up flags on an edge that lie on the same side. The involution σ pairs up flags on an edge that lie at the same endpoint. Finally, the involution ϕ pairs up neighboring flags of consecutive edges around a vertex. In this manner, an orbit of $\mathtt{Verts}_{\mathcal{E}} = \langle \sigma, \phi \rangle$ yields a cyclic order of flags around a vertex, an orbit of $\mathtt{Edges}_{\mathcal{E}} = \langle \theta, \sigma \rangle$ yields a cyclic order of the four flags of an edge, while an orbit of $\mathtt{Faces}_{\mathcal{E}} = \langle \theta, \phi \rangle$ yields a cyclic order of flags around a face. Note that our notion of a face slightly diverges from the topological one: Topologically, a single face can be bounded by several cycles, each composed of edges from a different connected component. In the graph embedding, we consider each such cycle to be a separate face. Nevertheless, due to the coefficient 2 of the term $|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})|$, the formula (2) corresponds to the standard notion of the Euler genus of an embedding.

In the other direction, note that a graph embedding $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ induces naturally a graph with vertex set being the set of orbits of $\mathtt{Verts}_{\mathcal{E}} = \langle \sigma, \phi \rangle$, edge set being the set of orbits of $\mathtt{Edges}_{\mathcal{E}} = \langle \theta, \sigma \rangle$ and edge being incident to a vertex if the corresponding two orbits share a flag. Observe that such an induced graph does not contain any isolated vertices. In other words, our notion of a graph embeddings ignores isolated vertices.

The book of Mohar and Thomassen [14] and the algorithms for finding low-genus embeddings used in the proof of Theorem 1.3 [9,13] operate on so-called *rotation systems* that consists of a cyclic order of endpoints of edges around every vertex and a signature $\lambda : E(G) \to \{\pm 1\}$. Intuitively, while going along an edge e with $\lambda(e) = -1$, one "flips" the roles of left and right. Using the intuition in the previous paragraph,

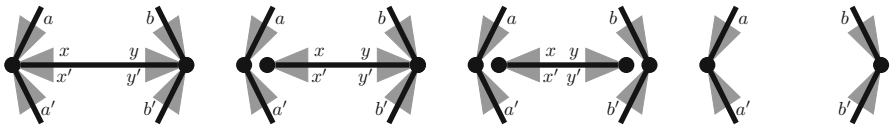

Fig. 2 Subsequent steps of edge deletion, assuming that all the 8 named flags are distinct

it is straightforward to transfer a rotation system into a graph embedding as defined in this section: one splits every edge into four flags, two for each endpoint, define σ and ϕ such that the orbits of $\langle \sigma, \phi \rangle$ correspond to the cyclic order of the endpoints in the rotation system, and finally adjust θ to reflect, for every edge e, if going along this edge keeps the directions of the orbits (when $\lambda(e) = 1$) or reverses them (when $\lambda(e) = -1$).

3.2 Basic Operations on Embeddings

We will need an operation of *deleting an edge* and a reverse operation of *drawing a new edge*.

3.2.1 Deleting an Edge

Let $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ be an embedding and let e be an edge of $\mathcal{E}$ with flags

$$x,\ x' = \sigma(x),\ y = \theta(x),\ y' = \sigma(y) = \theta(x').$$

We define an embedding $\mathcal{E} - e$, the result of deletion of the edge e from $\mathcal{E}$, in the following manner; see Fig. 2.

First, we disconnect the edge e from the rest of the embedding. For this, we modify ϕ so that $\{x, x'\}$ and $\{y, y'\}$ become its orbits: We set $a = \phi(x)$ and $a' = \phi(x')$, and we replace $\{x, a\}$ and $\{x', a'\}$ with $\{x, x'\}$ and $\{a, a'\}$. (Note that if $a = x'$ or $a' = x$, then this operation is void, because $\{x, a\} = \{x', a'\} = \{a, a'\} = \{x, x'\}$.) Next, we set $b = \phi(y)$ and $b' = \phi(y')$, and we replace $\{y, b\}$ and $\{y', b'\}$ with $\{y, y'\}$ and $\{b, b'\}$. As the third and final step, we delete the flags of e and the corresponding orbits of all three involutions.

Note that the first two steps cannot be conveyed in parallel because $\phi(y)$ and $\phi(y')$ might be altered during the first step. However, this happens only if e is a loop (with $a \in e$ or $a' \in e$), and a direct check shows that $\{x, x'\}$ and $\{y, y'\}$ both become orbits of ϕ and that ϕ keeps being an involution. What is more, the order of the first two steps (i.e., the arbitrary decision of processing $\{x, x'\}$ prior to $\{y, y'\}$) is irrelevant and edge deletion results in deleting the flags of e from the cycles corresponding to the vertices incident to e. This interpretation supports the following observation.

Observation 3.6 Let $\mathcal{E}_0 = \mathcal{E} - e$ for an embedding $\mathcal{E}$ and an edge e. For every vertex v_0 of $\mathcal{E}_0$, there exists a distinct vertex v of $\mathcal{E}$ such that the cycle of v_0 is the cycle of v with the flags of e removed. In the other direction, for every vertex v of $\mathcal{E}$, either all flags of v are contained in e, or there exists a vertex v_0 of $\mathcal{E}_0$ with the cycle equal to the cycle of v with the flags of e removed.

Next, we describe how genus changes subject to edge deletion.

Lemma 3.7 *Let e be an edge in the embedding $\mathcal{E}$, and let $\mathcal{E}_0 = \mathcal{E} - e$. Then the genus of $\mathcal{E}_0$ is not larger than the genus of $\mathcal{E}$. Furthermore, if e is incident to two faces or to a size-2 vertex, then the genera of $\mathcal{E}_0$ and $\mathcal{E}$ are equal.*

Proof Let $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$, $\mathcal{E}_0 = (\mathbf{F}_0, \theta_0, \sigma_0, \phi_0)$, and let x, x', y, y' be the four flags of e (aligned as in Fig. 2). We now investigate how the number of edges, vertices, faces, and connected components can change while deleting an edge e. Observe that this operation only affects orbits intersecting e. In particular, this yields $|\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}_0})| = |\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})| - 1$. To analyze orbits of the remaining types, we consider several cases.

Case A The edge e is incident to two distinct faces. Let f and f' be the faces incident to e, with cycles x, y, P and x', y', P', respectively, for some (possibly empty) sequences of flags P and P'.

Case A.1 Both faces f and f' are of size 2. In this case, both P and P' are empty, so f and f' get deleted. Moreover, the flags of e form a single vertex (with cycle x, y, y', x') and a separate connected component; these orbits are deleted as well. Thus,

$$|\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}_0})| = 2, \quad |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}_0})| = 1,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| = 1.$$

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) = 0$.

Case A.2. One of the faces f or f' is of size at least 4. Without loss of generality, suppose that f is of size at least 4, i.e., P is not empty. Observe that P remains in $\mathcal{E}_0$ as a sequence of flags connecting $\phi(x)$ with $\phi(y)$, where every two consecutive flags form an orbit either of ϕ_0 or of θ_0. Consequently, neither vertex incident to e is deleted, and these vertices still belong to the same connected component. Finally, observe that the deletion of e replaces the faces f and f' with one face, whose cycle is $P\bar{P}'$, where $\bar{P}'$ is the sequence P' reversed. Thus,

$$|\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}_0})| = 1, \quad |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}_0})| = 0,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| = 0.$$

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) = 0$.

Case B The edge e is incident to one face. Let f be the face incident to e. Moreover, let v and v' be vertices incident to e, with $x, x' \in v$ and $y, y' \in v'$; note that these vertices may coincide. We consider several cases based on the sizes of v and v'.

Case B.1 Both v and v' are of size 2. In this case, both v and v' get deleted along with e. Moreover, observe that the flags of e form a single face (with cycle x, y, y', x') and a separate connected component; these orbits are deleted as well. Thus,

$$|\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}_0})| = 1, \quad |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}_0})| = 2,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| = 1.$$

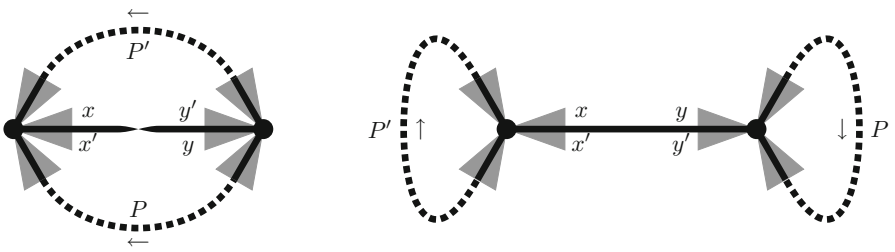

Fig. 3 An edge e incident to a single face f and two distinct vertices of size 6 or more. The walk around f may change the orientation between visiting both sides of e (to the left) or preserve it (to the right)

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) = 0$.

Case B.2 Exactly one of the vertices v and v' is of size 2. Without loss of generality, suppose that v' is of size 2, whereas v contains flags $\phi(x)$ and $\phi(x')$ which do not belong to e. In this case, v' gets deleted, while v is preserved. Moreover, v is still connected to all other vertices in the connected component of v (except for the deleted vertex v'), so the number of connected components does not change. Finally, observe that the cycle of f is x, y, y', x', P for some non-empty sequence of flags P, which starts with $\phi(x')$ and ends with $\phi(x)$. After the deletion, P forms a separate face. Thus,

$$|\mathsf{orb}(\texttt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\texttt{Faces}_{\mathcal{E}_0})| = 0, \quad |\mathsf{orb}(\texttt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\texttt{Verts}_{\mathcal{E}_0})| = 1,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| = 0.$$

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) = 0$.

Case B.3 The vertex $v = v'$ is of size 4. In this case, the flags of e form a single vertex (with cycle x, x', y, y'), a single face (with cycle x, y, x', y'), and a separate connected component. Thus,

$$|\mathsf{orb}(\texttt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\texttt{Faces}_{\mathcal{E}_0})| = 1, \quad |\mathsf{orb}(\texttt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\texttt{Verts}_{\mathcal{E}_0})| = 1,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| = 1.$$

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) = 1$.

Case B.4 Neither v nor v' is contained in e. In this case, the vertices v and v' (which may coincide) do not get deleted. The change in the number of faces depends on the cycle of f; see Fig. 3 for an illustration.

If the cycle of f is x, y, P, x', y', P' (for sequences of flags P and P', at least one of which is non-empty), the face f gets transformed to a face with cycle $P\bar{P}'$, where $\bar{P}'$ is the reverse of P'. Moreover, this face provides connectivity between the vertices v and v'. Hence, neither the number of faces nor the number of connected components changes. Thus,

$$|\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}_0})| = 0, \quad |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}_0})| = 0,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| = 0.$$

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) = 1$.

On the other hand, if the cycle of f is x, y, P, y', x', P' (for non-empty sequences of flags P and P'), then the deletion of e replaces f with two faces, whose cycles are P and P', respectively; hence, the number of faces increases by 1. The vertices v and v' may get disconnected or stay in the same connected component. Thus,

$$|\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}_0})| = -1, \quad |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}_0})| = 0,$$
$$|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| - |\mathsf{orb}(\mathsf{cc}_{\mathcal{E}_0})| \in \{0, -1\}.$$

Consequently, $\hat{g}(\mathcal{E}) - \hat{g}(\mathcal{E}_0) \in \{0, 2\}$.

We conclude the proof with a simple observation that the case distinction is complete. □

A direct consequence of Lemma 3.7 is the following.

Corollary 3.8 *The genus of an embedding is always non-negative.*

Proof Observe that any embedding can be turned into an empty embedding (i.e., one with no flags) by successive edge deletions. Since an edge deletion cannot increase the genus (Lemma 3.7), and the empty embedding has genus zero, the claim follows. □

3.2.2 Drawing a New Edge

Let us now define a reverse operation to edge deletion. Before, let us introduce a notion of *position* $p_{\mathcal{E}}(x)$ of a flag x. Let $x, y = \theta(x), y' = \sigma(y), x' = \sigma(x)$ be the flags contained in the edge e containing x. We set

$$p_{\mathcal{E}}(x) = \begin{cases} (\phi(x), \phi(y)) & \text{if } \phi(y) \notin \{x, x'\}, \\ (\phi(x), \phi(x')) & \text{if } \phi(y) = x, \\ (\phi(x), \phi(x)) & \text{if } \phi(y) = x'. \end{cases}$$

It is easy to observe that defining the process of deletion of e, we set a and b so that $(a, b) = p_{\mathcal{E}}(x)$. This lets us use $p_{\mathcal{E}}(x)$ to undo the deletion. While defining the operation of drawing a new edge, we want to avoid the need to give names to the flags of this newly added edge. Hence, to specify its position (a, b), we write $a = \perp$ instead of $a = x'$, $a = \top$ instead of $a = y$, $a = \top'$ instead of $a = y'$, and $b = \perp$ instead of $b = y'$.

Let $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ be an embedding, and let $a \in \mathbf{F} \cup \{\perp, \top, \top'\}$ and $b \in \mathbf{F} \cup \{\perp\}$. By *drawing a new edge at* (a, b) we mean the following operation, resulting in an embedding $\mathcal{E}_0$; see Fig. 4. First, we add four new flags to $\mathbf{F}$, denoted x, x', y, y' henceforth. We set $\{x, x'\}$ and $\{y, y'\}$ to be new orbits of σ and ϕ, whereas $\{x, y\}$ and $\{x', y'\}$ to be two new orbits of θ. Next, we replace $a = \perp$, $a = \top$, $a = \top'$, and

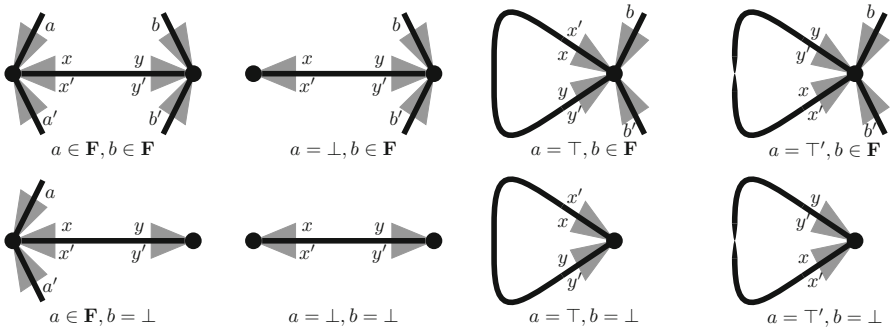

Fig. 4 The result of drawing an edge (x, y, y', x') at (a, b) for different combinations of a and b. Each case is illustrated with one out of potentially many configurations. In particular, $a, b \in \mathbf{F}$ does not imply $a' \in \mathbf{F}$

$b = \bot$ with $a = x'$, $a = y$, $a = y'$, and $b = y'$, respectively. Finally, we adjust ϕ in the following two steps: We define $b' = \phi(b)$ and replace the orbits $\{y, y'\}$ and $\{b, b'\}$ with $\{y, b\}$ and $\{y', b'\}$. (Note that if $b = y'$, then this operation is void because $\{y, y'\} = \{b, b'\} = \{y, b\} = \{y', b'\}$.) Then, we define $a' = \phi(a)$ and replace the orbits $\{x, x'\}$ and $\{a, a'\}$ with $\{x, a\}$ and $\{x', a'\}$. (Observe that this operation is void if $a = x'$.)

We defined this operation so that it is clear that if $e = (x, y, y', x')$ is an edge of $\mathcal{E}$, then one can retrieve $\mathcal{E}$ from $\mathcal{E} - e$ by drawing a new edge at $p_{\mathcal{E}}(x)$ and appropriately identifying the four new flags with x, x', y, y'. It is also easy to verify that drawing a new edge is always well-defined and that if $\mathcal{E}_0$ is obtained from $\mathcal{E}$ by drawing a new edge e, then $\mathcal{E} = \mathcal{E}_0 - e$.

Note that $a = \bot$ and $b = \bot$ both result in creating a new vertex (two new vertices if $a = \bot$ and $b = \bot$ hold simultaneously). Moreover, if $a = \bot$, then e is incident to a size-2 vertex (x, x'), if $a = \top$, then e is a loop incident to two faces including a size-2 face (x, y), while if $a = \top'$, then e is a loop incident to one face.

We identify one more special case of drawing a new edge. If a and b are distinct flags that lie on the same face f, and furthermore, the order of flags on the cycle of f is $a, P, b, \phi(b), P', \phi(a)$ for some (possibly empty) sequences of flags P and P', then we say that the new edge drawn at (a, b) is *drawn along the boundary of* f. Observe that if this is the case, then the new edge e is incident to two faces: in the new embedding $\mathcal{E}_0$, the face f has been split into a face with cycle a, P, b, y, x and a face with cycle $\phi(b), P', \phi(a), x', y'$. We explicitly allow here also the case $b = \phi(a)$; then the cycle of f is a, P, b for some sequence P, and the embedding $\mathcal{E}_0$ has new faces with cycles a, P, b, y, x and x', y', respectively.

Consequently, from Lemma 3.7 we immediately obtain the following.

Lemma 3.9 *If $\mathcal{E}_0$ is created from $\mathcal{E}$ by drawing a new edge, then the genus of $\mathcal{E}_0$ is not smaller than the genus of $\mathcal{E}$. Furthermore, the genera of $\mathcal{E}$ and $\mathcal{E}_0$ are equal if the edge has been drawn along a face boundary or at (a, b) with $a \in \{\bot, \top\}$.*

3.3 *t*-boundaried Embeddings

Definition 3.10 (*t-boundaried embedding*) A *t-boundaried embedding* is a tuple $(\mathcal{E}, t, L)$, where $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ is an embedding, t is a non-negative integer, and L is an injective function from a subset of $\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})$ to $[t]$. The elements of the domain of L are called *labelled vertices*, and the elements of $[t]$ are *labels*.

The genus of a t-boundaried embedding $(\mathcal{E}, t, L)$ is the genus of the underlying embedding $\mathcal{E}$.

The main motivation to introduce t-boundaried embeddings is to then merge them.

Definition 3.11 (*Merge of two t-boundaried embeddings*) Let $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$ be two t-boundaried embeddings, where $\widehat{\mathcal{E}}_i = (\mathcal{E}_i, t, L_i)$ and $\mathcal{E}_i = (\mathbf{F}_i, \theta_i, \sigma_i, \phi_i)$ for $i = 1, 2$, and the sets of flags $\mathbf{F}_1$ and $\mathbf{F}_2$ are disjoint. A t-boundaried embedding $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ with $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ is a *merge* of $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$ if it can be created by the following process.

First, we take $\widehat{\mathcal{E}}$ to be a disjoint union of $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$, that is, we take:

$$\mathbf{F} = \mathbf{F}_1 \cup \mathbf{F}_2 \qquad \theta = \theta_1 \cup \theta_2$$
$$\sigma = \sigma_1 \cup \sigma_2 \qquad \phi = \phi_1 \cup \phi_2$$
$$L = L_1 \cup L_2$$

Then, for every label $\ell \in [t]$ that is contained in the image of both L_1 and L_2 that is, there exists a cycle C_i corresponding to the orbit $L_i^{-1}(\ell)$ for $i = 1, 2$, we modify ϕ on the elements of C_1 and C_2 so that these elements form a single orbit of $\langle \sigma, \phi \rangle$ whose cycle C is a merge of C_1 and C_2. (Note that the cycle C is, in general, *not* defined uniquely.) We assign this cycle the label ℓ in the assignment L.

Observe that, as we only modify ϕ in the merge operation, the set of edges of a merge is the union of the edges of the components. In particular, it follows that every edge of a merge consists of four flags, and thus the merge is indeed a (graph) embedding.

For two t-boundaried embeddings $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$, we denote the family of all merges of $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$ as $M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)$. Moreover, as $\hat{g}_M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)$ we denote the minimum genus of a merge of $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$. Formally, $\hat{g}_M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) = \min\{\hat{g}(\widehat{\mathcal{E}}) : \widehat{\mathcal{E}} \in M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)\}$. A merge $\widehat{\mathcal{E}} \in M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)$ attaining this genus is called a *genus-minimum* merge.

Definition 3.12 (*Equivalence of t-boundaried embeddings*) Two t-boundaried embeddings $\widehat{\mathcal{E}}$ and $\widehat{\mathcal{E}}'$ are *equivalent* if for every t-boundaried embedding $\widehat{\mathcal{E}}_0$, the genera $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$ and $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}')$ are equal.

In a certain sense, merging is a commutative and associative operation:

Observation 3.13 Let $\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2, \widehat{\mathcal{E}}_3$ be arbitrary t-boundaried embeddings with pairwise disjoint flags, and let $\widehat{\mathcal{E}}_\emptyset$ be the empty t-boundaried embedding. The function $M(\cdot, \cdot)$ satisfies the following properties:

1. $\bigcup\{M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_{23}) : \widehat{\mathcal{E}}_{23} \in M(\widehat{\mathcal{E}}_2, \widehat{\mathcal{E}}_3)\} = \bigcup\{M(\widehat{\mathcal{E}}_{12}, \widehat{\mathcal{E}}_3) : \widehat{\mathcal{E}}_{12} \in M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)\}$.
2. $M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) = M(\widehat{\mathcal{E}}_2, \widehat{\mathcal{E}}_1)$,
3. $M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_\emptyset) = \{\widehat{\mathcal{E}}_1\}$,

Motivated by the associativity, we generalize the functions $M()$ and $\hat{g}_M()$ to an arbitrary finite arity.

3.3.1 Basic Operations on t-Boundaried Embeddings

By Observation 3.6, there is a natural correspondence between the vertices of $\mathcal{E}$ and $\mathcal{E} - e$ for every edge e of $\mathcal{E}$. This correspondence allows us to extend the definition of edge deletion to t-boundaried embeddings: If $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ and e is an edge in $\mathcal{E}$, then a t-boundaried embedding $\widehat{\mathcal{E}} - e$ is defined as $(\mathcal{E} - e, t, L_0)$, where for every vertex v_0 of $\mathcal{E} - e$, the labelling L_0 copies the label of the corresponding vertex v of $\mathcal{E}$. Note that the range of L_0 may be a proper subset of the range of L if some labelled vertex of $\widehat{\mathcal{E}}$ has all its flags contained in the deleted edge e. The characterization of vertex cycles in Observation 3.6 lets us relate edge deletion to merging.

Observation 3.14 Consider a merge $\widehat{\mathcal{E}}$ of t-boundaried embeddings $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$. For every edge e of $\widehat{\mathcal{E}}_1$, we have that $\widehat{\mathcal{E}} - e$ is a merge of $\widehat{\mathcal{E}}_1 - e$ and $\widehat{\mathcal{E}}_2$.

Similarly, we can define the operation of drawing a new edge in a t-boundaried embedding; if a new vertex is created by this operation (due to a or b being equal to $\perp$), we need to specify its label (or the fact that it is unlabelled). Thus, for each label ℓ unused in $\widehat{\mathcal{E}}$, we add a special value $\perp_\ell$ available for a and b, denoting the fact that the new vertex is labelled ℓ; the ordinary $\perp$ denotes the fact that it is unlabelled. Unlike Observation 3.14, its counterpart for drawing an edge requires dealing with a special situation.

Observation 3.15 Consider a merge $\widehat{\mathcal{E}}$ of t-boundaried embeddings $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$, and let $\widehat{\mathcal{E}}'_1$ be an embedding obtained from $\widehat{\mathcal{E}}_1$ by drawing an edge at (a, b). Then a merge of $\widehat{\mathcal{E}}'_1$ and $\widehat{\mathcal{E}}_2$ can be obtained from $\widehat{\mathcal{E}}$ by drawing an edge at $(\tilde{a}, \tilde{b})$, where $\tilde{a} = a$ and $\tilde{b} = b$ except for the following situation: if $a = \perp_\ell$ (or $b = \perp_\ell$) for a label ℓ used in $\widehat{\mathcal{E}}$ but not in $\widehat{\mathcal{E}}_1$, then $\tilde{a}$ (resp. $\tilde{b}$) is an arbitrary flag of $\widehat{\mathcal{E}}$ contained in a vertex with label ℓ.

3.4 Nice Embeddings

We say that a vertex or a face is *isolated* if its set of flags forms a separate connected component.

Definition 3.16 (*Nice (t-boundaried) embedding*) A t-boundaried embedding $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ with $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$ is *nice* if the following two conditions hold:

1. If an unlabelled vertex consists of less than 6 flags, then it is isolated. Equivalently, size-2 unlabelled vertices are forbidden, while each size-4 unlabelled vertex must be incident to a loop.

2. If an edge e is incident to two faces, then for every face f incident to e, if we denote by x and $y = \theta(x)$ the two flags contained both in f and e, then there is a flag $z \in f \setminus \{x, y, \phi(x), \phi(y)\}$ contained in a labelled vertex.

Our goal in this section is to show that any embedding can be turned into an equivalent nice one. The main motivation for such a cleaning step is that a nice embedding enjoys a good size bound due to the Euler formula-style estimations, presented in the next section.

3.4.1 Nice Embeddings are Small

Lemma 3.17 *A t-boundaried nice embedding $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ of genus $\hat{g}(\widehat{\mathcal{E}})$ satisfies*

$$|\mathbf{F}(\mathcal{E})| \leq 48t + 24\hat{g}(\widehat{\mathcal{E}}).$$

Proof Let $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$. We perform a discharging argument. The setup is as follows:

- every labelled vertex receives a charge of 2;
- every isolated vertex receives a charge of 1;
- every isolated face receives a charge of 1;
- every edge receives a charge of $\frac{5}{6}$.

The total initial charge is at most

$$2t + 2|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| + \tfrac{5}{6}|\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})|.$$

Then, we move the charge according to the following rules:

1. Every labelled vertex that is incident to only one face sends a charge of 1 to the face it is incident to.
2. Every edge that is incident only to labelled vertices divides a charge of $\frac{2}{3}$ equally among the faces it is incident to. In other words, every flag in such an edge sends a charge of $\frac{1}{6}$ to the face it is contained in.
3. Every edge that is incident to two vertices, one labelled and one unlabelled, gives a charge of $\frac{1}{3}$ to the unlabelled incident vertex and divides the remaining charge of $\frac{1}{2}$ equally between the faces it is incident to. In other words, in the first step every flag in such an edge contained in an unlabelled vertex sends a charge of $\frac{1}{6}$ to the vertex it is contained in. In the second step, every flag in such an edge sends a charge of $\frac{1}{8}$ to the face it is contained in.
4. Every edge that is incident only to unlabelled vertices divides a charge of $\frac{2}{3}$ equally among the vertices it is incident to. In other words, every flag in such an edge sends a charge of $\frac{1}{6}$ to the vertex it is contained in.

Clearly, every edge is left with a non-negative charge (0 or $\frac{1}{6}$). We now show that at the end of the process, every vertex and every face has a charge of at least one.

First, let us consider a vertex v. If v is labelled or isolated, 1 out of its initial charge remained, and the claim is straightforward. Next, we assume that v is unlabelled and

not isolated. By the first property of a nice embedding, v is of size at least 6. It received exactly $\frac{1}{6}$ from each of its flags, i.e., at least 1 in total.

Consider now a face f. We make a case distinction depending on how many flags of f belong to labelled vertices and how these flags are located on the cycle corresponding to f.

No flags of f belong to a labelled vertex Observe that every edge e incident to f is incident only with one face, as otherwise it would contradict the second property of a nice embedding. Consequently, f is isolated and it received an initial charge of 1.

Exactly 2 flags of f belong to a labelled vertex Suppose f is incident to only one labelled vertex v and shares two flags x and $y = \phi(x)$ with v. Let e be the edge containing x. By the second property of a nice embedding, f is the only face incident to e. Consequently, $\sigma(x)$ belongs to f. Since $\sigma(x)$ also belongs to v, which is a labelled vertex, we infer that $y = \sigma(x)$, the vertex v is of size 2, and f is the only face v is incident to. Thus, f received a charge of 1 from v.

Exactly 4 flags of f belong to labelled vertices and they are consecutive along f. Let x, $y = \theta(x)$, $x' = \phi(x)$, and $y' = \phi(y)$ be these four flags, and let e be the edge containing x and y. If e is incident to two faces, then it violates the second property of a nice embedding. Otherwise, the orbit $\{\sigma(x), \sigma(y)\}$ of θ appears on f. Since only 4 flags of f belong to labelled vertices, we need to have $\{\sigma(x), \sigma(y)\} = \{x', y'\}$, so f is an isolated face of size 4. Hence, it received an initial charge of 1.

The face f consists of 6 flags and they all belong to labelled vertices In this case all flags in f are contained in edges incident only to labelled vertices. Thus, f receives a charge of $\frac{1}{6}$ from each of these flags, which is 1 in total.

Along f there exist two nonconsecutive orbits of ϕ that are included in labelled vertices Let $\{x, x'\}$ and $\{y, y'\}$ be these orbits. Since they are nonconsecutive, $\theta(x), x, x', \theta(x'), \theta(y), y, y', \theta(y')$ are eight pairwise distinct flags in f. Each such flag belongs to an edge incident to at least one labelled vertex and thus sends a charge of at least $\frac{1}{8}$ to the face f.

Note that the last three cases cover the case of f having at least 4 flags contained in labelled vertices.

We have shown that there was enough charge so that every vertex and every face received a charge of at least one. Consequently,

$$|\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| + |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| \leq 2t + 2|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| + \tfrac{5}{6}|\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})|.$$

Together with (2), this implies

$$\begin{aligned}\hat{g}(\widehat{\mathcal{E}}) &= |\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Faces}_{\mathcal{E}})| - |\mathsf{orb}(\mathtt{Verts}_{\mathcal{E}})| + 2|\mathsf{orb}(\mathsf{cc}_{\mathcal{E}})| \\ &\geq \tfrac{1}{6}|\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})| - 2t,\end{aligned}$$

i.e.,

$$|\mathbf{F}| = 4|\mathsf{orb}(\mathtt{Edges}_{\mathcal{E}})| \leq 48t + 24\hat{g}(\widehat{\mathcal{E}}).$$

□

3.4.2 Making an Embedding Nice

Let $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ be a t-boundaried embedding with $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$. Our goal now is to obtain an equivalent nice embedding. To this end, we show that the following three operations lead to equivalent embeddings:

1. deleting an edge incident to an unlabelled size-2 vertex;
2. deleting an edge violating the second property of the definition of a nice embedding;
3. suppressing a size-4 unlabelled vertex that is not isolated.

We will henceforth call them *simplifying operations*.

Lemma 3.18 (Deleting an edge incident to a size-2 unlabelled vertex) *If e is an edge of $\widehat{\mathcal{E}}$ incident to an unlabelled vertex of size* 2*, then $\widehat{\mathcal{E}} - e$ and $\widehat{\mathcal{E}}$ are equivalent.*

Proof We consider merges of $\widehat{\mathcal{E}}$ and $\widehat{\mathcal{E}} - e$ with an arbitrary t-boundaried embedding $\widehat{\mathcal{E}}_0$.

First, let $\widehat{\mathcal{E}}_M$ be a genus-minimum merge of $\widehat{\mathcal{E}}_0$ and $\widehat{\mathcal{E}}$. Observation 3.14 yields $\widehat{\mathcal{E}}_M - e \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}} - e)$. Combined with Lemma 3.7, this implies $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}} - e) \leq \hat{g}(\widehat{\mathcal{E}}_M - e) \leq \hat{g}(\widehat{\mathcal{E}}_M) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$.

In the other direction, let $\widehat{\mathcal{E}}'_M$ be a genus-minimum merge of $\widehat{\mathcal{E}} - e$ and $\widehat{\mathcal{E}}_0$. Let $e = (x, y, y', x')$, where $\{x, x'\}$ is an unlabelled size-2 vertex so that $p_{\widehat{\mathcal{E}}}(x)$ is of the form (x', b). Consequently, $\widehat{\mathcal{E}}$ can be obtained from $\widehat{\mathcal{E}} - e$ by drawing a new edge e at $(\perp, b)$ for some b. By Observation 3.15, a merge $\widehat{\mathcal{E}}_M \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$ can be obtained from $\widehat{\mathcal{E}}'_M$ by drawing a new edge at $(\tilde{a}, \tilde{b})$. Since $a = \perp$, we have $\tilde{a} = \perp$, so the genera of $\widehat{\mathcal{E}}_M$ and $\widehat{\mathcal{E}}'_M$ are equal due to Lemma 3.9. Hence, $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}) \leq \hat{g}(\widehat{\mathcal{E}}_M) = \hat{g}(\widehat{\mathcal{E}}'_M) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}} - e)$. □

Lemma 3.19 (Deleting an edge violating the second property of the definition of nice embedding) *Let e be an edge in a t-boundaried embedding $\widehat{\mathcal{E}}$ that is incident to two faces. Furthermore, assume that some face f incident to e has the following property: if x and $y = \theta(x)$ are the two flags shared between e and f, then each flag $z \notin \{x, \phi(x), y, \phi(y)\}$ of f belongs to an unlabelled vertex. Then $\widehat{\mathcal{E}}$ is equivalent with $\widehat{\mathcal{E}} - e$.*

Proof Again, we consider merges of $\widehat{\mathcal{E}}$ and $\widehat{\mathcal{E}} - e$ with an arbitrary t-boundaried embedding $\widehat{\mathcal{E}}_0$.

In one direction, the proof is the same as in the proof of the previous lemma: Observation 3.14 and Lemma 3.7 imply $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}} - e) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$.

In the other direction, let $\widehat{\mathcal{E}}'_M$ be a genus-minimum merge of $\widehat{\mathcal{E}}_0$ and $\widehat{\mathcal{E}} - e$. We consider two cases depending on whether f is of size 2.

If so, then $p_{\widehat{\mathcal{E}}}(x)$ is of the form (y, b), and $\widehat{\mathcal{E}}$ can be obtained from $\widehat{\mathcal{E}} - e$ by drawing a new edge at $(\top, b)$ for some b. By Observation 3.15, a merge $\widehat{\mathcal{E}}_M \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$ can be obtained from $\widehat{\mathcal{E}}'_M$ by drawing a new edge at $(\tilde{a}, \tilde{b})$. Since $a = \top$, we have $\tilde{a} = \top$, so the genera of $\widehat{\mathcal{E}}_M$ and $\widehat{\mathcal{E}}'_M$ are equal due to Lemma 3.9.

Next, suppose that f contains a flag $z \notin \{x, y\}$. Since e is incident to 2 faces, we conclude that $\phi(x)$ and $\phi(y)$ do not belong to e, and $\widehat{\mathcal{E}}$ can be obtained from $\widehat{\mathcal{E}} - e$ by drawing a new edge at (a, b) for $a = \phi(x)$ and $b = \phi(y)$. By Observation 3.15, a merge $\widehat{\mathcal{E}}_M \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$ can be obtained from $\widehat{\mathcal{E}}'_M$ by drawing a new edge at (a, b). Let the cycle of f be x, a, P, b, y for some (possibly empty) sequence of flags P. By the assumptions of the lemma, every flag of P belongs to an unlabelled vertex. Consequently, in $\widehat{\mathcal{E}}'_M$ there exists a face f' whose cycle contains consecutive flags a, P, b on its cycle, as no orbit of ϕ or θ on a, P, b has been altered. This means that a new edge drawn at (a, b) is actually drawn along a face boundary. Hence, the genera of $\widehat{\mathcal{E}}_M$ and $\widehat{\mathcal{E}}'_M$ are equal due to Lemma 3.9.

In both cases, we obtain $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}) \leq \hat{g}(\widehat{\mathcal{E}}_M) = \hat{g}(\widehat{\mathcal{E}}'_M) = \hat{g}_M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}} - e)$. □

For the last basic operation, we need to formally define it. Let $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ be a t-boundaried embedding with an unlabelled size-4 vertex v that is not isolated. Since v is not isolated, v is incident to two edges e_1 and e_2, and each e_i is incident to v and a vertex $v_i \neq v$. Note that it is possible that $v_1 = v_2$.

Let x_1 and $x_2 = \phi(x_1)$ be two flags in v such that x_i belongs to e_i. Furthermore, let $y_i = \theta(x_i)$; note that y_i lies in e_i. We delete all four flags of v and replace the orbits of θ on e_1 and e_2 with $\{y_1, y_2\}$ and $\{\sigma(y_1), \sigma(y_2)\}$. In other words, we replace e_1 and e_2 with a new edge with flags $y_1, y_2, \sigma(y_1), \sigma(y_2)$. We now formally verify that the output embedding is an equivalent one.

Lemma 3.20 (Suppressing a size-4 unlabelled vertex that is not isolated) *The operation of suppressing a size-4 unlabelled vertex leads to an equivalent embedding.*

Proof We interpret the suppressing operation as a sequence of one edge drawing and two edge deletions.

Observe that y_1, x_1, x_2, y_2 are four distinct flags that lie on the same face f and, furthermore, they are consecutive in this order along the cycle of f. Let us draw a new edge e_f along the boundary of f, at y_1 and y_2, obtaining an embedding $\widehat{\mathcal{E}}'$. Note that in $\widehat{\mathcal{E}}'$ there is a new face f' with cycle $(z_1, y_1, x_1, x_2, y_2, z_2)$, where $\{z_1, z_2\}$ is a new orbit of θ contained in the edge e_f. Furthermore, the edge e_f with face f' fulfills the assumptions of Lemma 3.19, and its deletion from $\widehat{\mathcal{E}}_f$ gives $\widehat{\mathcal{E}}$. Consequently, $\widehat{\mathcal{E}}'$ and $\widehat{\mathcal{E}}$ are equivalent.

Then, we delete edges e_1 and e_2 from $\widehat{\mathcal{E}}'$, obtaining an embedding $\widehat{\mathcal{E}}''$. Our goal is to show that such an operation leads to an equivalent embedding; note that if we rename the flags of e_f to $y_1, y_2, \sigma(y_1), \sigma(y_2)$, we obtain the output embedding of the suppressing operation. The proof of equivalence of $\widehat{\mathcal{E}}''$ and $\widehat{\mathcal{E}}'$ is similar to that of Lemma 3.19, but without most of the special cases due to the existence of the edge e_f.

Let $\widehat{\mathcal{E}}_0$ be an arbitrary t-boundaried embedding. By Observation 3.14 and Lemma 3.7, $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}'') \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}')$.

For the other direction, let $\widehat{\mathcal{E}}''_M = (\mathcal{E}''_M, t, L''_M)$ be a genus-minimum merge of $\widehat{\mathcal{E}}_0$ and $\widehat{\mathcal{E}}''$. Note that $p_{\widehat{\mathcal{E}}'}(x_1) = (x_2, z_1)$ and $p_{\widehat{\mathcal{E}}'-e_1}(x_2) = (x'_2, z_2)$. Hence, $\widehat{\mathcal{E}}'$ can be retrieved from $\widehat{\mathcal{E}}''$ by drawing edge e_2 at $(\perp, z_2)$ and then edge e_1 at (x_2, z_1). By Observation 3.15, a merge $\widehat{\mathcal{E}}'_M \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}')$ can be retrieved from $\widehat{\mathcal{E}}''_M$ by drawing an edge e_2 at $(\perp, z_2)$ and then e_1 at (x_2, z_1). Lemma 3.9 asserts that the first of these operations preserves the genus. As for the second operation, we observe that after drawing e_2 flags x_2, y_2, z_2, z_1 form a fragment of a face boundary and that e_1 is drawn along it. Hence, Lemma 3.9 also implies that the genera of $\widehat{\mathcal{E}}''_M$ and $\widehat{\mathcal{E}}'_M$ are equal. We infer $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}') \leq \hat{g}(\widehat{\mathcal{E}}'_M) = \hat{g}(\widehat{\mathcal{E}}''_M) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}'')$, which concludes the proof of the lemma. □

Armed with the three basic operations, we are now ready to prove the following statement.

Lemma 3.21 *There exists a polynomial-time algorithm that, given an arbitrary t-boundaried embedding, computes an equivalent nice one by successive applications of the simplifying operations.*

Proof Let $\widehat{\mathcal{E}}$ be a t-boundaried embedding. If $\widehat{\mathcal{E}}$ is nice, we can just return $\widehat{\mathcal{E}}$. Otherwise, observe that any object that violates the niceness of $\widehat{\mathcal{E}}$ gives rise to an applicability of one of the simplifying operations. An edge violating the second property of a nice embedding can be deleted from $\widehat{\mathcal{E}}$; the resulting embedding is equivalent due to Lemma 3.19. Similarly, Lemma 3.18 lets us safely delete the edge incident to an unlabelled size-2 vertex, and a size-4 unlabelled vertex that is not isolated can be suppressed due to Lemma 3.20.

Finally, note that each execution of a simplifying operation strictly decreases the number of flags in the embedding. Consequently, in polynomial time we obtain an equivalent nice embedding. □

Due to Lemma 3.17, this algorithm can be applied to transform a t-boundaried embedding into an equivalent one with flags from a small universe.

Corollary 3.22 *There exists a function $U(t, g) = 48t + 24g$ and a polynomial-time algorithm that, given a universe $\mathbf{U}$ and a t-boundaried embedding $\widehat{\mathcal{E}}$ such that $|\mathbf{U}| \geq U(t, \hat{g}(\widehat{\mathcal{E}}))$, constructs an equivalent t-boundaried embedding $\widehat{\mathcal{E}}'$ whose flags $\mathbf{F}(\widehat{\mathcal{E}}')$ form a subset of $\mathbf{U}$.*

4 Bounded Treewidth Graphs

In this section, we prove Theorem 1.2. Without loss of generality, we can assume that the input tree decomposition $(\mathbf{T}, \beta)$ of the input graph G is a nice tree decomposition of width *less* than t, that is, every bag of $(\mathbf{T}, \beta)$ is of size at most t.

We further refine $(\mathbf{T}, \beta)$ as follows. First, with every node $\mathbf{t}$ we associate a graph $G(\mathbf{t})$, initialized as $G(\mathbf{t}) := G^{\downarrow}(\mathbf{t})$. In the refinement process, we will maintain the invariant that at every node $\mathbf{t}$, the graph $G(\mathbf{t})$ is a subgraph of $G[\alpha(\mathbf{t})]$ and a supergraph of $G^{\downarrow}(\mathbf{t})$; in particular, $V(G(\mathbf{t})) = \alpha(\mathbf{t})$. Note that only $\mathcal{O}(t^2)$ bits are needed to keep $G(\mathbf{t})$ at every node $\mathbf{t}$ since it suffices to store which edges of $G[\beta(\mathbf{t})]$ belong to $G(\mathbf{t})$.

For every forget node $\mathbf{t}$, we perform the following refinement process. Let $\mathbf{t}'$ be the child of $\mathbf{t}$, and let v be the forgotten vertex, i.e., $\{v\} = \beta(\mathbf{t}')\setminus\beta(\mathbf{t})$. Let $e_1, e_2, \ldots, e_\ell$ be the edges incident to v that have their second endpoint in $\beta(\mathbf{t})$. Note that, since we do not allow loops in the input graph G, then $\{e_1, e_2, \ldots, e_\ell\}$ are exactly the edges that are present in $G^{\downarrow}(\mathbf{t})$ but are not present in $G^{\downarrow}(\mathbf{t}')$. Our goal is to refine the edge $\mathbf{tt}'$ of $\mathbf{T}$ by inserting a number of vertices on this edge so that the transition from $G^{\downarrow}(\mathbf{t}')$ to $G^{\downarrow}(\mathbf{t})$ is smoother.

More precisely, we subdivide the edge $\mathbf{tt}'$ ℓ times, inserting a path comprising vertices $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_\ell$ in this order, such that $\mathbf{t}_1$ becomes the child of $\mathbf{t}$, and $\mathbf{t}_\ell$ becomes the parent of $\mathbf{t}'$. Furthermore, to every node $\mathbf{t}_i$ we add a second child $\mathbf{s}_i$ that is a leaf of the tree $\mathbf{T}$. The new node $\mathbf{s}_i$ is of a new type, namely an **edge leaf**: we set $\beta(\mathbf{s}_i) = \beta(\mathbf{t}')$, $V(G(\mathbf{s}_i)) = \beta(\mathbf{t}')$, and $E(G(\mathbf{s}_i)) = \{e_i\}$. The node $\mathbf{t}_i$ is a **join** node with $\beta(\mathbf{t}_i) = \beta(\mathbf{t}')$ and $G(\mathbf{t}_i) = G^{\downarrow}(\mathbf{t}) - \{e_1, e_2, \ldots, e_{i-1}\}$.

This completes the description of the tree decomposition refinement. By this step, we have obtained the following properties: at every introduce or forget node $\mathbf{t}$ with child $\mathbf{t}'$, we have $E(G(\mathbf{t})) = E(G(\mathbf{t}'))$, while at every join node $\mathbf{t}$ with children $\mathbf{t}_1$ and $\mathbf{t}_2$ we have $E(G(\mathbf{t})) = E(G(\mathbf{t}_1)) \uplus E(G(\mathbf{t}_2))$. Furthermore, note that we have introduced two nodes for every edge of G, giving $\mathcal{O}(nt)$ new nodes in total.

We compute a labelling function $\Lambda : V(G) \to [t]$ such that Λ is injective on every bag of $(\mathbf{T}, \beta)$; such a labelling Λ is straightforward to construct in a top-to-bottom fashion using the fact that every bag of $(\mathbf{T}, \beta)$ is of size at most t.

With the refined tree decomposition, we perform a bottom-up dynamic programming algorithm. Fix a node $\mathbf{t} \in V(\mathbf{T})$. For a subset $A \subseteq \alpha(\mathbf{t})$ and an embedding $\mathcal{E}$ of $G(\mathbf{t})[A]$, we can naturally equip $\mathcal{E}$ with a structure of a t-boundaried embedding by assigning labels $\Lambda|_{\beta(\mathbf{t})}$ (which is an injective function). We will denote such a t-boundaried embedding at node $\mathbf{t}$ as $\mathsf{bnd}(\mathbf{t}, \mathcal{E})$. Intuitively, the bnd function extends an embedding $\mathcal{E}$ with a "boundary" induced by $\mathbf{t}$ (because $\beta(\mathbf{t})$ is a "boundary" of $G(\mathbf{t})$ as a subgraph of G).

At $\mathbf{t}$, we maintain a DP table with values $T[\mathbf{t}, X, \widehat{\mathcal{E}}] \in \{0, 1, \ldots\} \cup \{+\infty\}$ for every set $X \subseteq \beta(\mathbf{t})$ and for every t-boundaried embedding $\widehat{\mathcal{E}}$ whose labels belong to $\Lambda(\beta(\mathbf{t})\setminus X)$ and whose flags form a subset of a universe $\mathbf{U}(\mathbf{t})$. These universes are chosen to be of size $U(t, g) = \mathcal{O}(t + g)$ as defined in Corollary 3.22, and so that they are disjoint between leaves, edge leaves, and join nodes. The universe $\mathbf{U}(\mathbf{t})$ of a forget or an introduce node $\mathbf{t}$ is shared with the only child of $\mathbf{t}$.

Aiming for an intuitive semantics, we would like the value $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ to indicate the minimum size of a deletion set $Y \subseteq \alpha(\mathbf{t})\setminus\beta(\mathbf{t})$ such that $G(\mathbf{t}) - (X \cup Y)$ can be embedded equivalently with $\widehat{\mathcal{E}}$. However, this natural definition causes several technical issues, including the following two:

- We would need to make sure $T[\mathbf{t}, X, \widehat{\mathcal{E}}] = T[\mathbf{t}, X, \widehat{\mathcal{E}}']$ if the $\widehat{\mathcal{E}}$ and $\widehat{\mathcal{E}}'$ are equivalent; this seems to require a subroutine testing whether two t-boundaried embeddings are equivalent.
- Consider retrieval of the witness embedding of $G(\mathbf{t}) - (X \cup Y)$, where $\mathbf{t}$ is a join node with children $\mathbf{t}_1$ and $\mathbf{t}_2$. If $T[\mathbf{t}, X, \widehat{\mathcal{E}}] = T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_1] + T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_2]$ and $\widehat{\mathcal{E}}$ is equivalent to a merge $\widehat{\mathcal{E}}' \in M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)$, a natural strategy would be to merge the embeddings of $G(\mathbf{t}_i) - (X \cup Y_i)$ equivalent to $\widehat{\mathcal{E}}_i$ into an embedding

of $G(\mathbf{t}) - (X \cup Y_1 \cup Y_2)$ equivalent to $\widehat{\mathcal{E}}$; however, it is not clear whether such a merge exists.

While handling these issues seems to require much effort, they are mere artifacts of the natural yet very strong requirements on the values of the DP table. Hence, we reduce these conditions to the bare essentials and maintain only the following two properties:

(a finite value yields a small deletion set) If $T[\mathbf{t}, X, \widehat{\mathcal{E}}] \neq +\infty$, then for every t-boundaried embedding $\widehat{\mathcal{E}}_0$ with flags disjoint with $\mathbf{U}(\mathbf{t})$, there exists a set $Y \subseteq \alpha(\mathbf{t})\backslash\beta(\mathbf{t})$ of size at most $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ and an embedding $\mathcal{E}$ of $G(\mathbf{t}) - (X \cup Y)$ such that $\hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E})) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$.

(a good deletion set is represented in the table) For all sets $X \subseteq \beta(\mathbf{t})$ and $Y \subseteq \alpha(\mathbf{t})\backslash\beta(\mathbf{t})$, every embedding $\mathcal{E}$ of $G(\mathbf{t}) - (X \cup Y)$ of genus at most g, and every t-boundaried embedding $\widehat{\mathcal{E}}_0$ such that the flags of $\widehat{\mathcal{E}}_0$ are disjoint with both the flags of $\mathcal{E}$ and with $\mathbf{U}(\mathbf{t})$, there exists an entry $T[\mathbf{t}, X, \widehat{\mathcal{E}}] \leq |Y|$ such that $\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E}))$.

Compared to the intuitive semantics, the above invariants include only one inequality from the definition of equivalence. Moreover, the existential quantifiers are moved past the universal quantifiers so that the embeddings can be constructed based on $\widehat{\mathcal{E}}_0$. These two features let us avoid the aforementioned two issues.

We remark that in both properties one can restrict $\widehat{\mathcal{E}}_0$ to use only labels of $\Lambda(\beta(t)\backslash X)$, as only those labels actively participate in the merge operation.

Having weakened the invariants, we need to make sure that they are strong enough to be still useful for the Genus Vertex Deletion problem. More precisely, we shall prove for the root $\mathbf{r}$ of $\mathbf{T}$ that the minimum value $T[\mathbf{r}, \emptyset, \cdot]$ is the size of the minimum solution to Genus Vertex Deletion on (G, g). Recall that $\beta(\mathbf{r}) = \emptyset$ and $G(\mathbf{r}) = G$. By the second invariant, an optimum solution Y^* and the underlying embedding $\mathcal{E}^*$ of $G - Y^*$ when merged with the empty embedding $\widehat{\mathcal{E}}_\emptyset$ has its corresponding entry $T[\mathbf{r}, \emptyset, \widehat{\mathcal{E}}^*] \leq |Y^*|$. In the other direction, the first invariant ensures that for every entry $T[\mathbf{r}, \emptyset, \widehat{\mathcal{E}}]$ and again the empty embedding $\widehat{\mathcal{E}}_\emptyset$, there exists a deletion set Y of size at most $T[\mathbf{r}, \emptyset, \widehat{\mathcal{E}}]$ and an embedding $\mathcal{E}$ of $G - Y$ such that

$$\hat{g}(\mathcal{E}) = \hat{g}(\mathsf{bnd}(\mathbf{r}, \mathcal{E})) = \hat{g}_M(\widehat{\mathcal{E}}_\emptyset, \mathsf{bnd}(\mathbf{r}, \mathcal{E})) \leq \hat{g}_M(\widehat{\mathcal{E}}_\emptyset, \widehat{\mathcal{E}}) = \hat{g}(\widehat{\mathcal{E}}) \leq g.$$

In particular, Y is a solution to Genus Vertex Deletion on (G, g).

We now show how to compute the values $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ in a bottom-up fashion. The table is always initialized with values $+\infty$ and its entries are altered through a sequence of *updates*. We say that updating a cell $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ with a value a results in setting $T[\mathbf{t}, X, \widehat{\mathcal{E}}] := \min(a, T[\mathbf{t}, X, \widehat{\mathcal{E}}])$. An update is *successful* if the stored value changed.

4.1 Leaf Nodes

A leaf node $\mathbf{t}$ has an empty graph $G(\mathbf{t})$, which makes processing it very simple: we just construct the empty t-boundaried embedding $\widehat{\mathcal{E}}_\emptyset$ and update the cell $T[\mathbf{t}, \emptyset, \widehat{\mathcal{E}}_\emptyset]$ with a value 0. Note there are other cells $T[\mathbf{t}, \emptyset, \widehat{\mathcal{E}}]$ with $\widehat{\mathcal{E}}$ equivalent to the empty embedding, but we keep their initial values $+\infty$. As we prove below, our invariants are weak enough to allow for handling them this way.

First property Consider the only cell $T[\mathbf{t}, \emptyset, \widehat{\mathcal{E}_\emptyset}]$ with a finite value (equal to 0) and an arbitrary t-boundaried embedding $\widehat{\mathcal{E}_0}$. Taking $Y = \emptyset$ and the empty embedding $\mathcal{E}_\emptyset$ of the empty graph $G(\mathbf{t})$, we obtain $\hat{g}_M(\widehat{\mathcal{E}_0}, \mathsf{bnd}(\mathbf{t}, \mathcal{E}_\emptyset)) = \hat{g}_M(\widehat{\mathcal{E}_0}, \widehat{\mathcal{E}_\emptyset})$ due to $\mathsf{bnd}(\mathbf{t}, \mathcal{E}_\emptyset) = \widehat{\mathcal{E}_\emptyset}$.

Second property Since $G(\mathbf{t})$ is empty, it suffices to consider $X = \emptyset$, $Y = \emptyset$, the empty embedding $\mathcal{E}_\emptyset$ of the empty graph, and an arbitrary t-boundaried embedding $\widehat{\mathcal{E}_0}$. Observe the empty t-boundaried embedding $\widehat{\mathcal{E}_\emptyset} = \mathsf{bnd}(\mathbf{t}, \mathcal{E}_\emptyset)$ satisfies both $T[\mathbf{t}, \emptyset, \widehat{\mathcal{E}_\emptyset}] = 0 = |Y|$ and $\hat{g}_M(\widehat{\mathcal{E}_0}, \widehat{\mathcal{E}_\emptyset}) = \hat{g}_M(\widehat{\mathcal{E}_0}, \mathsf{bnd}(\mathbf{t}, \mathcal{E}_\emptyset))$.

4.2 Edge Leaf Nodes

Recall that an edge leaf node $\mathbf{t}$ satisfies $\alpha(t) = \beta(t)$ and $|E(G(\mathbf{t}))| = 1$. Moreover, the only edge $\{v, v'\}$ of $G(\mathbf{t})$ is not a loop because we do not allow loops in the input graph G. For an edge leaf node $\mathbf{t}$, we construct two t-boundaried embeddings: an empty embedding $\widehat{\mathcal{E}_\emptyset}$ and an embedding $\widehat{\mathcal{E}}$ obtained from $\widehat{\mathcal{E}_\emptyset}$ by drawing an edge at $(\perp_{\Lambda(v)}, \perp_{\Lambda(v')})$ using flags from the universe $\mathbf{U}(\mathbf{t})$ associated with the edge leaf node in question. For each $X \subseteq \beta(\mathbf{t}) \setminus \{v, v'\}$, we update $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ with a value 0, while for the remaining subsets $X \subseteq \beta(\mathbf{t})$, we update $T[\mathbf{t}, X, \widehat{\mathcal{E}_\emptyset}]$ with a value 0. Note that in the first case, there are other cells $T[\mathbf{t}, \emptyset, \widehat{\mathcal{E}'}]$ with $\widehat{\mathcal{E}'}$ *isomorphic* to $\widehat{\mathcal{E}}$ (equal to $\widehat{\mathcal{E}}$ up to renaming flags), but we keep their values $+\infty$.

Observe that for every $X \subseteq \beta(\mathbf{t})$, the graph $G(\mathbf{t}) - X$ has (up to isomorphism) a unique embedding: an empty embedding $\mathcal{E}_\emptyset$ if $X \cap \{v, v'\} \neq \emptyset$ (when the graph has no edges) and a planar embedding $\mathcal{E}$ otherwise (when the graph has one edge). Moreover, $\mathsf{bnd}(\mathbf{t}, \mathcal{E}_\emptyset) = \widehat{\mathcal{E}_\emptyset}$ and $\mathsf{bnd}(\mathbf{t}, \mathcal{E})$ is isomorphic to $\widehat{\mathcal{E}}$. Consequently, one can easily prove (like for the leaf nodes) that the DP table $T[\mathbf{t}, \cdot, \cdot]$ satisfies the required properties.

4.3 Introduce Nodes

Let $\mathbf{t}$ be an introduce node with child $\mathbf{t}'$ and let v be the introduced vertex. Observe that $G(\mathbf{t})$ can be obtained from $G(\mathbf{t}')$ by introducing v as an isolated vertex. To handle $\mathbf{t}$, we iterate over finite cells $T[\mathbf{t}', X, \widehat{\mathcal{E}}]$ and copy their values to both $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ and $T[\mathbf{t}, X \cup \{v\}, \widehat{\mathcal{E}}]$. The notion of an embedding ignores isolated vertices, so the resulting table satisfies the required properties by the inductive assumption on the table of the child $\mathbf{t}'$.

4.4 Forget Nodes

Let $\mathbf{t}$ be a forget node with child $\mathbf{t}'$ and let v be the forgotten vertex. Note that, due to the refinement step, we have $G(\mathbf{t}') = G(\mathbf{t})$. The only difference between the nodes $\mathbf{t}$ and $\mathbf{t}'$ is that if we consider some embedding $\mathcal{E}$ of a subgraph of $G(\mathbf{t})$, then $\mathsf{bnd}(\mathbf{t}', \mathcal{E})$ assigns a label $\Lambda(v)$ to the vertex v, while in $\mathsf{bnd}(\mathbf{t}, \mathcal{E})$ the vertex v remains unlabelled.

Before we proceed, let us formally introduce the operation of *forgetting* a label. Given a t-boundaried embedding $\widehat{\mathcal{E}} = (\mathcal{E}, t, L)$ and a label $\ell \in [t]$, we say that $\widehat{\mathcal{E}}' = (\mathcal{E}, t, L')$ is obtained from $\widehat{\mathcal{E}}$ by *forgetting the label* ℓ if $L'(v)$ is undefined for $v \in L^{-1}(\ell)$ and $L'(v) = L(v)$ otherwise; note that $\widehat{\mathcal{E}}' = \widehat{\mathcal{E}}$ if ℓ is not in the range of L. The resulting t-boundaried embedding $\widehat{\mathcal{E}}'$ is denoted $\widehat{\mathcal{E}} \ominus \ell$. As observed above, for any embedding $\widehat{\mathcal{E}}$ of a subgraph of $G(\mathbf{t})$, we have $\mathsf{bnd}(\mathbf{t}, \mathcal{E}) = \mathsf{bnd}(\mathbf{t}', \mathcal{E}) \ominus \Lambda(v)$.

This behavior is straightforward to implement in our DP tables. For every cell $T[\mathbf{t}', X, \widehat{\mathcal{E}}]$ with $v \in X$, we update the cell $T[\mathbf{t}, X \setminus \{v\}, \widehat{\mathcal{E}}]$ with a value $T[\mathbf{t}', X, \widehat{\mathcal{E}}] + 1$: when v is deleted, the embedding does not change, but we need to account for the vertex v in the value of the cell. For every cell $T[\mathbf{t}', X, \widehat{\mathcal{E}}]$ with $v \notin X$, on the other hand, we update the cell $T[\mathbf{t}, X, \widehat{\mathcal{E}} \ominus \Lambda(v)]$ with a value $T[\mathbf{t}', X, \widehat{\mathcal{E}}]$.

Before we formally verify that the entries for node $\mathbf{t}$ satisfy the required properties, let us we explore the interplay between merging and forgetting a label.

Observation 4.1 Let $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$ be t-boundaried embeddings with disjoint flags and let $\ell \in [t]$. We have

$$\{\widehat{\mathcal{E}} \ominus \ell : \widehat{\mathcal{E}} \in M(\widehat{\mathcal{E}}_1 \ominus \ell, \widehat{\mathcal{E}}_2)\} = M(\widehat{\mathcal{E}}_1 \ominus \ell, \widehat{\mathcal{E}}_2 \ominus \ell) = \{\widehat{\mathcal{E}} \ominus \ell : \widehat{\mathcal{E}} \in M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2 \ominus \ell)\}.$$

Consequently,

$$\hat{g}_M(\widehat{\mathcal{E}}_1 \ominus \ell, \widehat{\mathcal{E}}_2) = \hat{g}_M(\widehat{\mathcal{E}}_1 \ominus \ell, \widehat{\mathcal{E}}_2 \ominus \ell) = \hat{g}_M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2 \ominus \ell).$$

First property Take a cell $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ with a finite value and a t-boundaried embedding $\widehat{\mathcal{E}}_0$. Let $T[\mathbf{t}', X', \widehat{\mathcal{E}}']$ be the last cell that caused an update in the value of $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$; observe that $X = X' \setminus \{v\}$ and $\widehat{\mathcal{E}} = \widehat{\mathcal{E}}' \ominus \Lambda(v)$.

We inductively apply the first property to the cell $T[\mathbf{t}', X', \widehat{\mathcal{E}}']$ for the embedding $\widehat{\mathcal{E}}'_0 := \widehat{\mathcal{E}}_0 \ominus \Lambda(v)$. This results in a set Y' of size at most $T[\mathbf{t}', X', \widehat{\mathcal{E}}']$ and an embedding $\mathcal{E}'$ of $G(\mathbf{t}') - (X' \cup Y')$ such that $\hat{g}_M(\widehat{\mathcal{E}}'_0, \mathsf{bnd}(\mathbf{t}', \mathcal{E}')) \leq \hat{g}_M(\widehat{\mathcal{E}}'_0, \widehat{\mathcal{E}}')$.

To prove the first property for $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$, we consider two cases. If $v \in X'$, we set $Y := Y' \cup \{v\}$ so that $X \cup Y = X' \cup Y'$ and $|Y| = |Y'| + 1 \leq T[\mathbf{t}', X', \widehat{\mathcal{E}}'] + 1 = T[\mathbf{t}, X, \widehat{\mathcal{E}}]$. Otherwise, we set $Y := Y'$, which also yields $X \cup Y = X' \cup Y'$ and $|Y| = |Y'| \leq T[\mathbf{t}', X', \widehat{\mathcal{E}}'] = T[\mathbf{t}, X, \widehat{\mathcal{E}}]$. In either case, we also take $\mathcal{E} := \mathcal{E}'$ so that Observation 4.1 yields

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E})) = \hat{g}_M(\widehat{\mathcal{E}}'_0, \mathsf{bnd}(\mathbf{t}', \mathcal{E}')) \leq \hat{g}_M(\widehat{\mathcal{E}}'_0, \widehat{\mathcal{E}}') = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}})$$

due to $\mathsf{bnd}(\mathbf{t}, \mathcal{E}) = \mathsf{bnd}(\mathbf{t}', \mathcal{E}') \ominus \Lambda(v)$, $\widehat{\mathcal{E}}'_0 = \widehat{\mathcal{E}}_0 \ominus \Lambda(v)$, and $\widehat{\mathcal{E}} = \widehat{\mathcal{E}}' \ominus \Lambda(v)$.

Second property Take sets X and Y as well as embeddings $\mathcal{E}$ and $\widehat{\mathcal{E}}_0$ as in the statement of the second property. If $v \in Y$, take $Y' := Y \setminus \{v\}$ and $X' := X \cup \{v\}$; otherwise, take $Y' := Y$ and $X' := X$. In both cases, we have $X \cup Y = X' \cup Y'$, $X' \subseteq \beta(\mathbf{t}')$, and $Y' \subseteq \alpha(\mathbf{t}') \setminus \beta(\mathbf{t}')$. Moreover, define $\mathcal{E}' := \mathcal{E}$ and $\widehat{\mathcal{E}}'_0 := \widehat{\mathcal{E}}_0 \ominus \Lambda(v)$. We inductively apply the second property for $\mathbf{t}'$, X', Y', $\mathcal{E}'$, and $\widehat{\mathcal{E}}'_0$ to obtain an entry $T[\mathbf{t}', X', \widehat{\mathcal{E}}'] \leq |Y'|$ such that $\hat{g}_M(\widehat{\mathcal{E}}'_0, \widehat{\mathcal{E}}') \leq \hat{g}_M(\widehat{\mathcal{E}}'_0, \mathsf{bnd}(\mathbf{t}', \mathcal{E}'))$. While processing the cell $T[\mathbf{t}', X', \widehat{\mathcal{E}}']$, the algorithm attempts an update on a cell $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ for $\widehat{\mathcal{E}} = \widehat{\mathcal{E}}' \ominus \Lambda(v)$. A direct check

shows that regardless of whether v belongs to Y or not, it follows that $T[\mathbf{t}, X, \widehat{\mathcal{E}}] \leq |Y|$. Moreover, Observation 4.1 yields

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}) = \hat{g}_M(\widehat{\mathcal{E}}'_0, \widehat{\mathcal{E}}') \leq \hat{g}_M(\widehat{\mathcal{E}}'_0, \mathsf{bnd}(\mathbf{t}', \mathcal{E}')) = \hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E}))$$

due to $\mathsf{bnd}(\mathbf{t}, \mathcal{E}) = \mathsf{bnd}(\mathbf{t}', \mathcal{E}') \ominus \Lambda(v)$, $\widehat{\mathcal{E}}'_0 = \widehat{\mathcal{E}}_0 \ominus \Lambda(v)$, and $\widehat{\mathcal{E}} = \widehat{\mathcal{E}}' \ominus \Lambda(v)$. Thus, $\widehat{\mathcal{E}}$ fulfills the second property for X, Y, $\mathcal{E}$, and $\widehat{\mathcal{E}}_0$.

This completes the description and the proof of correctness of the computations at forget nodes.

4.5 Join Nodes

Let $\mathbf{t}$ be a join node with children $\mathbf{t}_1$ and $\mathbf{t}_2$. Note that we have $V(G(\mathbf{t}_1)) \cap V(G(\mathbf{t}_2)) = \beta(\mathbf{t})$, $V(G(\mathbf{t}_1)) \cup V(G(\mathbf{t}_2)) = V(G(\mathbf{t}))$ and $E(G(\mathbf{t})) = E(G(\mathbf{t}_1)) \uplus E(G(\mathbf{t}_2))$.

To compute the table $T[\mathbf{t}, \cdot, \cdot]$, we iterate over sets $X \subseteq \beta(\mathbf{t})$. For each X, we construct all t-boundaried embeddings $\widehat{\mathcal{E}}'$ whose flags form a subset of $\mathbf{U}(\mathbf{t}_1) \cup \mathbf{U}(\mathbf{t}_2)$. We discard embeddings with $\hat{g}(\widehat{\mathcal{E}}') > g$ and those whose labels do not belong to $\Lambda(\beta(\mathbf{t}) \setminus X)$. For each remaining embedding $\widehat{\mathcal{E}}'$, we construct two t-boundaried embeddings $\widehat{\mathcal{E}}_i$ (for $i \in \{1, 2\}$) by deleting all the edges except those contained in $\mathbf{U}(\mathbf{t}_i)$. We check if $\widehat{\mathcal{E}}'$ is a merge of $\widehat{\mathcal{E}}_1$ with $\widehat{\mathcal{E}}_2$ and discard it otherwise. We also discard $\widehat{\mathcal{E}}'$ if $T[\mathbf{t}_i, X, \widehat{\mathcal{E}}_i] = +\infty$ for some i. If $\widehat{\mathcal{E}}'$ has not been discarded, we apply Corollary 3.22 to construct an equivalent embedding $\widehat{\mathcal{E}}$ whose flags belong to $\mathbf{U}(\mathbf{t})$, and we update $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ with $T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_1] + T[\mathbf{t}_2, X, \widehat{\mathcal{E}}_2]$.

Observe that for each pair $(X, \widehat{\mathcal{E}}')$, the procedure described above takes polynomial time. The number of sets X is $2^{|\beta(\mathbf{t})|} \leq 2^t$, while the number of considered t-boundaried embeddings $\widehat{\mathcal{E}}'$ is $2^{\mathcal{O}(u \log u)}$, where $u = |\mathbf{U}(\mathbf{t}_1) \cup \mathbf{U}(\mathbf{t}_2)| = 2U(t, g) = \mathcal{O}(t + g)$. Consequently, the overall processing time of the node $\mathbf{t}$ is $2^{\mathcal{O}((t+g)\log(t+g))}$.

We now formally verify that the entries for node $\mathbf{t}$ satisfy the required properties.

First property Take a cell $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ with a finite value and a t-boundaried embedding $\widehat{\mathcal{E}}_0$. Assume the value of $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ comes from considering cells $T[\mathbf{t}_i, X, \widehat{\mathcal{E}}_i]$ and a merge $\widehat{\mathcal{E}}'$ of $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$. By the equivalence of $\widehat{\mathcal{E}}$ and $\widehat{\mathcal{E}}'$, we have

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}') = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}).$$

By Observation 3.13, a genus-minimum merge of $\widehat{\mathcal{E}}_0$, $\widehat{\mathcal{E}}_1$, and $\widehat{\mathcal{E}}_2$ can be obtained by merging $\widehat{\mathcal{E}}_{01} \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1)$ with $\widehat{\mathcal{E}}_2$. We apply the first property to the cell $T[\mathbf{t}_2, X, \widehat{\mathcal{E}}_2]$ with $\widehat{\mathcal{E}}_{01}$, obtaining a set Y_2 and an embedding $\mathcal{E}_2$ of $G(\mathbf{t}_2) - (X \cup Y_2)$ such that $|Y_2| \leq T[\mathbf{t}_2, X, \widehat{\mathcal{E}}_2]$ and $\hat{g}_M(\widehat{\mathcal{E}}_{01}, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) \leq \hat{g}_M(\widehat{\mathcal{E}}_{01}, \widehat{\mathcal{E}}_2)$. Consequently,

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) \leq \hat{g}_M(\widehat{\mathcal{E}}_{01}, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) \leq \hat{g}_M(\widehat{\mathcal{E}}_{01}, \widehat{\mathcal{E}}_2) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2).$$

By Observation 3.13, a genus-minimum merge of $\widehat{\mathcal{E}}_0$, $\widehat{\mathcal{E}}_1$, and $\mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)$ can be obtained by merging $\widehat{\mathcal{E}}_{02} \in M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2))$ with $\widehat{\mathcal{E}}_1$. We apply the first property to the cell $T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_1]$ with $\widehat{\mathcal{E}}_{02}$, obtaining a set Y_1 and an embedding $\mathcal{E}_1$ of $G(\mathbf{t}_1) - (X \cup Y_1)$ such that $|Y_1| \leq T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_1]$ and $\hat{g}_M(\widehat{\mathcal{E}}_{02}, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1)) \leq \hat{g}_M(\widehat{\mathcal{E}}_{02}, \widehat{\mathcal{E}}_1)$.

Consequently,

$$\begin{aligned}\hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1), \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) &\leq \hat{g}_M(\widehat{\mathcal{E}}_{02}, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1)) \leq \hat{g}_M(\widehat{\mathcal{E}}_{02}, \widehat{\mathcal{E}}_1) \\ &= \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)).\end{aligned}$$

By Observation 3.13, a genus-minimum merge of $\widehat{\mathcal{E}}_0$, $\mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1)$, and $\mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)$ can be obtained by merging $\widehat{\mathcal{E}}_{12} \in M(\mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1), \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2))$ with $\widehat{\mathcal{E}}_0$. Let $\mathcal{E}$ be the embedding underlying $\widehat{\mathcal{E}}_{12}$ and let $Y = Y_1 \cup Y_2$. Then, $\mathcal{E}$ is an embedding of $G(\mathbf{t}) - (X \cup Y)$ such that $\widehat{\mathcal{E}}_{12} = \mathsf{bnd}(\mathbf{t}, \mathcal{E})$. Hence,

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E})) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_{12}) = \hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1), \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}),$$

and the proof of the first property is finished.

Second property Take sets X and Y as well as embeddings $\mathcal{E}$ and $\widehat{\mathcal{E}}_0$ as in the statement of the second property. For $i = 1, 2$, take $Y_i = Y \cap \alpha(\mathbf{t}_i)$ and $\mathcal{E}_i$ to be the embedding $\mathcal{E}$ restricted to the edges of $G(\mathbf{t}_i)$ (i.e., with the edges of $G(\mathbf{t}_{3-i})$ deleted). Note that $\mathsf{bnd}(\mathbf{t}, \mathcal{E}) \in M(\mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1), \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2))$ and that each $\mathcal{E}_i$ is an embedding of $G(\mathbf{t}_i) - (X \cup Y_i)$ of genus at most g. Hence,

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1), \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E})).$$

By Observation 3.13, a genus-minimum merge of $\widehat{\mathcal{E}}_0$, $\mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1)$, and $\mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)$ can be obtained by merging $\widehat{\mathcal{E}}_{02} \in M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2))$ with $\mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1)$. We apply the second property to $\mathbf{t}_1$, X, Y_1, $\mathcal{E}_1$, and $\widehat{\mathcal{E}}_{02}$, obtaining an entry $T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_1] \leq |Y_1|$ such that $\hat{g}_M(\widehat{\mathcal{E}}_{02}, \widehat{\mathcal{E}}_1) \leq \hat{g}_M(\widehat{\mathcal{E}}_{02}, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1))$. Consequently,

$$\begin{aligned}\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) &\leq \hat{g}_M(\widehat{\mathcal{E}}_{02}, \widehat{\mathcal{E}}_1) \leq \hat{g}_M(\widehat{\mathcal{E}}_{02}, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1)) \\ &= \hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}_1, \mathcal{E}_1), \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)).\end{aligned}$$

By Observation 3.13, a genus-minimum merge of $\widehat{\mathcal{E}}_0$, $\widehat{\mathcal{E}}_1$, and $\mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)$ can be obtained by merging $\widehat{\mathcal{E}}_{01} \in M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1)$ with $\mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)$. We apply the second property to $\mathbf{t}_2$, X, Y_2, $\mathcal{E}_2$, and $\widehat{\mathcal{E}}_{01}$, obtaining an entry $T[\mathbf{t}_2, X, \widehat{\mathcal{E}}_2] \leq |Y_2|$ such that $\hat{g}_M(\widehat{\mathcal{E}}_{01}, \widehat{\mathcal{E}}_2) \leq \hat{g}_M(\widehat{\mathcal{E}}_{01}, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2))$. Consequently,

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \leq \hat{g}_M(\widehat{\mathcal{E}}_{01}, \widehat{\mathcal{E}}_2) \leq \hat{g}_M(\widehat{\mathcal{E}}_{01}, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \mathsf{bnd}(\mathbf{t}_2, \mathcal{E}_2)).$$

By Observation 3.13, a genus-minimum merge of $\widehat{\mathcal{E}}_0$, $\widehat{\mathcal{E}}_1$, and $\widehat{\mathcal{E}}_2$ can be obtained by merging $\widehat{\mathcal{E}}_{12} \in M(\widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2)$ with $\widehat{\mathcal{E}}_0$. Observe that the algorithm constructs $\widehat{\mathcal{E}}_{12}$ as one of the t-boundaried embeddings over $\mathbf{U}(\mathbf{t}_1) \cup \mathbf{U}(\mathbf{t}_2)$. Since $\widehat{\mathcal{E}}_{12}$ is a merge of $\widehat{\mathcal{E}}_1$ and $\widehat{\mathcal{E}}_2$, it attempts an update on the cell $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ for $\widehat{\mathcal{E}}$ equivalent to $\widehat{\mathcal{E}}_{12}$ with the value $T[\mathbf{t}_1, X, \widehat{\mathcal{E}}_1] + T[\mathbf{t}_2, X, \widehat{\mathcal{E}}_2] \leq |Y_1| + |Y_2| \leq |Y|$. Since

$$\hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_{12}) = \hat{g}_M(\widehat{\mathcal{E}}_0, \widehat{\mathcal{E}}_1, \widehat{\mathcal{E}}_2) \leq \hat{g}_M(\widehat{\mathcal{E}}_0, \mathsf{bnd}(\mathbf{t}, \mathcal{E})),$$

the entry $T[\mathbf{t}, X, \widehat{\mathcal{E}}]$ fulfills the second property for X, Y, $\mathcal{E}$, and $\widehat{\mathcal{E}}_0$.

4.6 Summary

We have concluded the descriptions and correctness proofs of the operations at the nodes of the tree decomposition. To analyze the running time, we observe that $|V(\mathbf{T})| = \mathcal{O}(nt)$ and that at every node $\mathbf{t}$, our algorithm takes $2^{\mathcal{O}((t+g)\log(t+g))}$ time to fill the DP table. This completes the proof of Theorem 1.2.

5 Irrelevant Vertex

In this section, we prove Theorem 1.3:

Theorem 1.3 *There exists a sequence $(C_g)_{g\geq 0}$ of positive integers and an algorithm that, given a graph G, non-negative integers g and k, and a set $M \subseteq V(G)$ such that $G - M$ is embeddable into a surface of Euler genus at most g, in time $C_g n^{\mathcal{O}(1)}$ finds one of the following:*

1. *a tree decomposition of G of width at most $C_g|M|^{1/2}k^{3/2}$; or*
2. *a vertex $w \in V(G)$ such that every solution to the* Genus Vertex Deletion *instance (G, g, k) contains w; or*
3. *a vertex $v \in V(G)$ such that for every $S \subseteq V(G)\setminus\{v\}$ the set S is a solution to the* Genus Vertex Deletion *instance (G, g, k) if and only if S is a solution to a* Genus Vertex Deletion *instance $(G - \{v\}, g, k)$.*

The argumentation is heavily inspired by the corresponding planar case by Marx and Schlotter [12]. As in most irrelevant vertex arguments, we follow the typical outline:

1. Run an approximation algorithm for treewidth and, in case it returns that the treewidth of the graph is larger than the required threshold, find a large grid minor. Here, a linear dependency on the grid size and treewidth in bounded genus graphs is known [4], and one can find the corresponding grid minor efficiently.
2. Show that a vertex $w \in M$ that is connected to many places in the grid that are far apart needs to be included in every solution, exactly as it is in the planar case. In the absence of such a vertex, a large part of the grid minor is flat, that is, embeds planarly and does not have any internal connections to the modulator M.
3. Prove that a middle vertex of such a flat part is irrelevant. Here, the challenge is to argue that in every solution there exists an embedding of the remaining part that draws the flat part indeed in a flat manner.

We use the following two auxiliary results on Euler genus throughout this section:

Lemma 5.1 ([5, Lemma B.6]) *Let Γ be a surface of Euler genus g and let $\mathcal{C}$ be a set of $g+1$ disjoint circles in Γ. If $\Gamma \setminus \bigcup \mathcal{C}$ has a connected component D_0 whose closure*

in Γ *meets every circle in* $\mathcal{C}$*, then at least one of the circles in* $\mathcal{C}$ *bounds a disc in* Γ *that is disjoint from* D_0.

Lemma 5.2 (Stahl and Beineke [16]) *The Euler genus of the graph* G *equals the sum of the Euler genera of the blocks (the 2-connected components) of* G.

Let G, k, g, and M be such as in the statement of Theorem 1.3. We start by computing an embedding $\mathcal{E}_0$ of $G - M$ into a surface of Euler genus at most g, using either the algorithm of Kawarabayashi et al. [9] or the older algorithm of Mohar [13].[2] This takes time $C_g^1 n$ for some constant C_g^1 depending only on g.

The algorithm represents the embedding as a (hyper)graph embedding defined in Sect. 3, that is, as $\mathcal{E} = (\mathbf{F}, \theta, \sigma, \phi)$. As discussed in Sect. 3, it is straightforward to treat this embedding as a rotation system as defined by Mohar and Thommassen [14]. In particular, it allows us to project $\mathcal{E}$ down on any subgraph of G by omitting the flags not from the subgraph in any orbit of $\mathsf{cc}_{\mathcal{E}} = \langle \theta, \sigma, \phi \rangle$, $\mathtt{Verts}_{\mathcal{E}} := \langle \sigma, \phi \rangle$, $\mathtt{Edges}_{\mathcal{E}} := \langle \theta, \sigma \rangle$, or $\mathtt{Faces}_{\mathcal{E}} := \langle \theta, \phi \rangle$.

5.1 Finding a Large Grid

We apply constant-factor approximation algorithms for treewidth [3, Theorem 3.10] and largest excluded grid minor [3, Theorem 3.11] in graphs with a fixed excluded minor. The algorithms do not need to compute a near-embedding of G because $\mathcal{E}_0$ serves this purpose. As described in [3, Theorems 3.10 and 3.11], the algorithms therefore run in time $C_g^2 n^{\mathcal{O}(1)}$ and are C_g^3-approximations, where C_g^2 and C_g^3 depend only on g.

If the resulting tree decomposition is of width at most $C_g^4 |M|^{1/2} k^{3/2}$ for some sufficiently large constant C_g^4, then we directly return it. Otherwise, by choosing the constant C_g^4 appropriately, we get a grid minor of side-length $\Theta(|M|^{1/2}(k+g)^{3/2} g^{1/2})$ as the dependency of treewidth and the largest grid minor in graphs excluding a fixed minor is linear [4].

A *wall* of side-length ℓ (see Fig. 5 for an illustration) consists of:

- $\ell + 1$ paths $P_0, P_1, \ldots, P_\ell$, where each path P_i is of length $2\ell + 1$, with vertices $v_{i,j}$, $0 \le j \le 2\ell + 1$ in this order, except for the path P_0 that is one edge shorter and does not contain the vertex $v_{0,2\ell+1}$ and the path P_ℓ that is one edge shorter and does not contain the vertex $v_{\ell,0}$ if ℓ is even and the vertex $v_{\ell,2\ell+1}$ if ℓ is odd;
- edges $v_{i,j} v_{i+1,j}$ for every pair $(i, j) \in \{0, 1, \ldots, \ell - 1\} \times \{0, 1, \ldots, 2\ell + 1\}$ of integers of the same parity.

Note that, in our definition, a wall of side-length ℓ has a natural planar embedding (as in Fig. 5) and the infinite face is surrounded by a simple cycle, which we call the *surrounding cycle* of the wall. Furthermore, the vertex $v_{\lfloor \ell/2 \rfloor, \ell}$ is called the *middle vertex* of the wall.

[2] The work [9] appeared so far only as an extended abstract in conference proceedings.

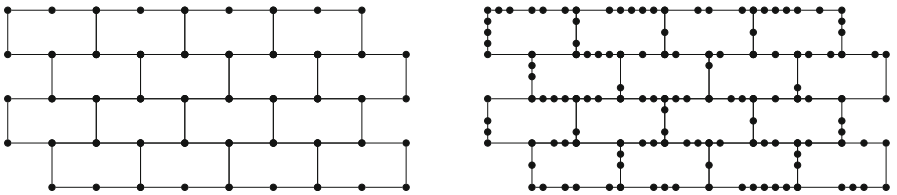

Fig. 5 A wall of side-length 4 and a subdivision of this wall

Clearly, a grid of side-length ℓ contains a wall of side-length $\lfloor \ell/2 \rfloor$ as a subgraph. Furthermore, a wall has maximum degree three. Consequently, if a graph contains a grid of side-length ℓ as a minor, it also contains a subdivision of a wall of side-length $\lfloor \ell/2 \rfloor$ as a subgraph. This allows us to henceforth focus on the case when in $G - M$ we get a subdivision W_0 of a wall $W_0^{\square}$ of side-length ℓ_0, where $\ell_0 \geq c_5|M|^{1/2}(k+g)^{3/2}g^{1/2}$ and c_5 can be chosen arbitrarily large.

In the wall $W_0^{\square}$, we identify $g+1$ pairwise disjoint subwalls of side-length $\ell_1 = \Theta(\ell_0/\sqrt{g}) \geq c_6|M|^{1/2}(k+g)^{3/2}$ (where the constant c_6 can be chosen arbitrarily large by choosing c_5 large as well), leaving enough space between the subwalls so that the part of $W_0^{\square}$ not contained in any of the subwalls is connected. By Lemma 5.1, for at least one of the identified subwalls, its surrounding cycle, when projected to W_0 in $G - M$, bounds a disc in the considered embedding of $G - M$. Consequently, the natural projection onto $G - M$ of at least one of the identified subwalls is planarly embedded in $\mathcal{E}_0$. We denote this subwall as $W_1^{\square}$, its projection in $G - M$ (being a subdivided wall) as W_1, and the part of $G - M$ embedded in the closed disc separated by the surrounding cycle of W_1 by G_1 (in particular, the surrounding cycle of W_1 is included in G_1).

In what follows, we mostly focus on G_1 and W_1. Without loss of generality, we henceforth assume that G_1 is connected: G_1 can be disconnected only if $G - M$ is disconnected, and a connected component of $G - M$ is planarly drawn inside one of the faces of the connected component of G_1 that contains W_1. If this is the case, we move the drawings of all such planar components into another face of the embedding, one not in the disc bounded by the surrounding cycle of W_1.

Observe that a wall admits only one planar embedding with its surrounding cycle as the boundary of the infinite face. Consequently, the embedding $\mathcal{E}_0$, restricted to W_1, is the natural embedding of a subdivision of a wall as depicted in Fig. 5. In what follows, we will analyze many wall-like subgraphs of W_1; the above observation implies that all these subgraphs inherit this natural embedding.

5.2 Filtering out the Modulator Neighbors

We now show that if a vertex $w \in M$ is adjacent to many scattered vertices of G_1, then it needs to be included in every solution. Recall that the boundary cycle of W_1 encloses a disc containing W_1 in the considered embedding $\mathcal{E}_0$ of $G - M$, and G_1 is the part of the graph $G - M$ that is enclosed by this boundary cycle. As discussed earlier, the embedding $\mathcal{E}_0$, restricted to W_1, is the natural embedding of a wall as depicted in Fig. 5. This allows us to introduce the following definition.

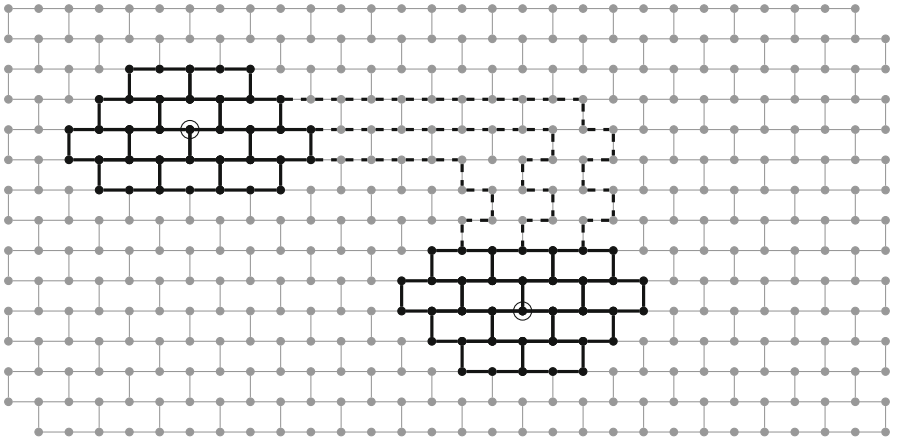

Fig. 6 A part of the subdivided wall W_1 (gray) with balls $B(v, 2)$ highlighted in black for two vertices $v \in I(w)$ surrounded by circles. The dashed paths is an example of a family $\mathcal{P}(\cdot, \cdot)$ consisting of three paths

For a vertex $v \in V(G_1)$ and an integer ℓ, the *radius-ℓ ball* $B(v, \ell)$ around v is defined as follows: we take all faces f of the embedding of the subdivided wall W_1 in $\mathcal{E}_0$, except for the infinite face, that contain v either inside or on the boundary, mark all faces that lie in face-vertex distance (in the embedding of W_1 in $\mathcal{E}_0$, but excluding traversal through the infinite face) at most ℓ from one of these faces f and put into $B(v, \ell)$ all vertices of G_1 that are contained inside or on the boundary of marked faces. Observe that from the fact that G_1 is connected it follows that every ball $B(v, \ell)$ induces a connected subgraph of G_1.

For every vertex $w \in M$, we create a set $I(w) \subseteq V(G_1)$ as follows. We start with $I(w) = \emptyset$ and every vertex of $V(G_1)$ unmarked (the marking scheme is performed from scratch for distinct vertices $w \in M$). As long as there exists an unmarked neighbor v of w in G_1, we insert v into $I(w)$ and mark all vertices of $B(v, \ell_2)$ for $\ell_2 = c_2(k + g + 1)$, where c_2 is an integer constant to be fixed later. We claim that if $I(w)$ is too large, the vertex w needs to be deleted in any solution to Genus Vertex Deletion.

Lemma 5.3 *Let $w \in M$ be a vertex for which $|I(w)| > k + g + 1$. Then w belongs to every solution to the* Genus Vertex Deletion *instance (G, g, k).*

Proof Suppose the contrary, and let S be a solution that does not contain w. That is, $|S| \leq k$, and $G - S$ admits an embedding $\mathcal{E}$ into a surface of Euler genus at most g.

Consider subgraphs $B(v, \ell_2/4)$ for $v \in I(w)$. By the construction of $I(w)$, these subgraphs are vertex-disjoint. Furthermore, as $\ell_2/4$ is much larger than k, for every $v_1, v_2 \in I(w)$, $v_1 \neq v_2$, there exists a set $\mathcal{P}(v_1, v_2)$ of $k+1$ vertex-disjoint paths in G_1 connecting the outer boundary of $B(v_1, \ell_2/4)$ with the outer boundary of $B(v_2, \ell_2/4)$, without any internal vertex in any of the subgraphs $B(v, \ell_2/4)$ for $v \in I(w)$; see Fig. 6. Let I' be the set of these vertices $v \in I(w)$ such that S is disjoint with $V(B(v, \ell_2/4))$; since $|S| \leq k$, we have $|I'| > g + 1$.

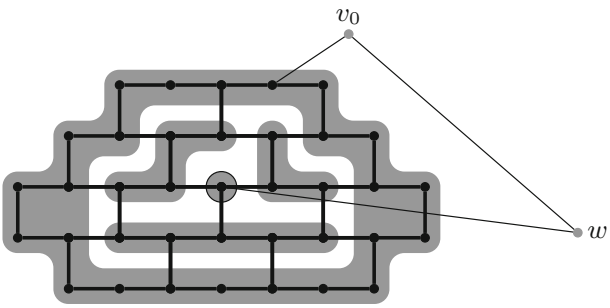

Fig. 7 Illustration how to contract a ball $B(v, 2)$ as in Fig. 6 into a K_5^- as in the proof of Lemma 5.3

Construct a minor H of $G - S$ as follows. Pick some arbitrary $v_0 \in I'$ and contract $B(v_0, \ell_2/4)$ onto v_0. For every $v \in I' \setminus \{v_0\}$, contract onto v_0 a path of $\mathcal{P}(v_0, v)$ that is disjoint with S. For every $v \in I' \setminus \{v_0\}$, contract $B(v, \ell_2/4)$ into a K_5^- (a five-vertex clique with one edge missing; recall that $B(v, \ell_2/4)$ contains a large wall, being part of W_1) such that one of the two nonadjacent vertices is adjacent to w and the other to v_0; see Fig. 7 for an illustration. Finally, contract all edges incident with w, including the edge wv_0. Observe that we have obtained a graph isomorphic to $|I'| - 1$ copies of a five-vertex clique K_5 (one for each $B(v, \ell_2/4)$ for $v \in I' \setminus \{v_0\}$) with one vertex from each clique identified with w. Since K_5 is not planar, by Lemma 5.2, the Euler genus of $G - S$ is at least $|I'| - 1 > g$, a contradiction. □

Consequently, if there exists $w \in M$ with $|I(w)| > k + g + 1$, then we can return w as the second result of the algorithm of Theorem 1.3. Henceforth, we will assume that $|I(w)| \leq k + g + 1$ for every $w \in M$.

Recall that W_1 is a subdivision of the wall $W_1^{\square}$ of side-length ℓ_1, where $\ell_1 \geq c_6|M|^{1/2}(k + g)^{3/2}$ and c_6 can be chosen arbitrarily large. This lets us identify n_2 disjoint subwalls of side-length $\ell_2 = c_2(k + g + 1)$, where $n_2 \geq c_7|M|(k + g)$ and c_7 can be chosen arbitrarily large. Note that $|\bigcup_{w \in M} I(w)| = \mathcal{O}(|M|(k + g))$ and that all vertices of $N_G(M) \cap V(G_1)$ are located in $\bigcup_{w \in M, v \in I(w)} B(v, \ell_2)$. Each ball $B(v, \ell_2)$ may intersect only a constant number of the identified subwalls of side-length ℓ_2. Hence, if c_7 is sufficiently large, there is a subdivision W_2 of a wall $W_2^{\square}$ of side-length $\ell_2 = c_2(k + g + 1)$ such that no vertex of a graph G_2, defined as the part of G_1 that is enclosed by the surrounding cycle of W_2, is a neighbor of a vertex in M. Note that only the vertices on the surrounding cycle of G_2 may have neighbors in $G - V(G_2)$. With G_2 and W_2, we proceed to the next section.

5.3 Middle Vertex of a Flat Part is Irrelevant

Let v be the vertex of W_2 that corresponds to the middle vertex of $W_2^{\square}$. We show that v is irrelevant, that is, it can be returned as the third outcome of Theorem 1.3. Clearly, if S is a solution to (G, g, k), then $S \setminus \{v\}$ is a solution to $(G - \{v\}, g, k)$. In the other direction, suppose that there exists $S \subseteq V(G) \setminus \{v\}$ of size at most k such

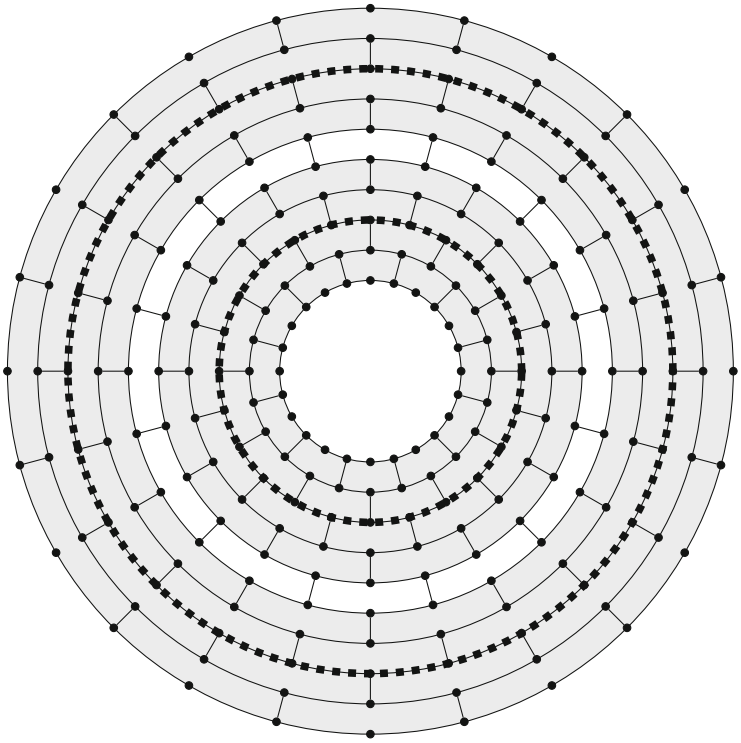

Fig. 8 A circular wall of height 9 and circumference 12 containing two rings. Their central circles are marked with black squares and the faces belonging to the territories of the rings are shaded in gray

that $G - \{v\} - S$ admits an embedding $\mathcal{E}$ into a surface of Euler genus at most g. We would like to enhance this embedding so that it also accommodates v.

Recall that the side-length ℓ_2 of $W_2^{\square}$ equals $c_2(k+g+1)$ for a constant c_2 that we can choose arbitrarily large. Observe that if c_2 is large enough, then we can find in W_2 a subdivision W_3 of a *circular wall* of height $h_3 = 5(k+g)+9$ and circumference $\ell_3 = \max(3, k+1)$ so that v is located inside the inner cycle of the wall; see Fig. 8 for the definition of a circular wall. A way how to find a circular wall inside a regular one is depicted in Fig. 9; the crucial observation is that a wall of side-length h contains $\Omega(h)$ concentric cycles with two consecutive cycles connected by a matching of size $\Omega(h)$, and the endpoints of the consecutive matchings interleaved as required.

A circular wall of height h contains $h+1$ naturally defined concentric cycles that we enumerate with integers $0, 1, \ldots, h$, starting from the inside. Next, in W_3 we identify $k+g+2$ *rings* (which are subdivisions of concentric circular walls of height 4 and circumference ℓ_3) that are vertex-disjoint, that is, the ith ring ($0 \leq i \leq k+g+1$) uses $(5i)$th up to $(5i+4)$th concentric cycle of the wall W_3; see Fig. 8. Let $\mathcal{R}_0$ be the family of identified rings.

For a ring $R \in \mathcal{R}_0$, we say that the *central circle* of R is the 2nd circle (i.e., the one between the two middle layers of the ring), the *inner boundary circle* is the 0th circle and the *outer boundary circle* is the 4th circle.

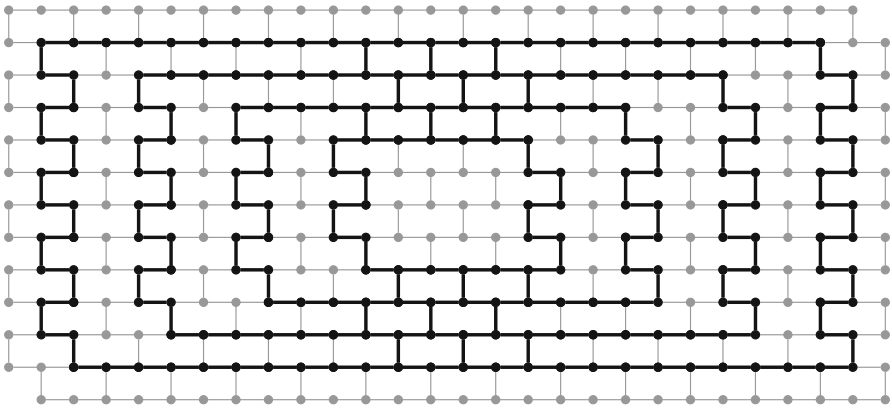

Fig. 9 How to find a subdivision of a circular wall inside a regular one

Recall that the embedding $\mathcal{E}_0$, restricted to W_1, is the natural embedding of a subdivision of a wall as depicted in Fig. 5. Since W_3 is a subgraph of W_1, the embedding $\mathcal{E}_0$ embeds W_3 and all rings of $\mathcal{R}_0$ as in Fig. 8. This allows us to introduce the following definitions.

Let $R \in \mathcal{R}_0$. A face of R in the embedding $\mathcal{E}_0$ restricted to R is *small* if it is not the outer face nor the face inside the 0th (innermost) cycle, and *central* if it is incident to the 2nd (central) circle. The *territory* of R, denoted by $T(R)$, is the subgraph of G that consists of R and everything that is drawn in the embedding $\mathcal{E}_0$ in the small faces of R. Note that $T(R)$ is a planar graph, and the subgraphs $\{T(R) : R \in \mathcal{R}_0\}$ are vertex-disjoint.

Let $\mathcal{R} \subseteq \mathcal{R}_0$ be the family of these rings for which $T(R)$ is disjoint with S. As $|\mathcal{R}_0| = k + g + 2$ and the subgraphs $T(R)$ are vertex-disjoint, we have $|\mathcal{R}| \geq g + 2$.

The previously defined inside-to-outside order of the concentric cycles in W_3 imposes a natural linear order on the rings of $\mathcal{R}_0$ and the corresponding induced order of $\mathcal{R}$. This allows us to speak about two *consecutive* rings of $\mathcal{R}$ or $\mathcal{R}_0$.

We say that a ring $R \in \mathcal{R}$ is embedded *plainly* in the embedding $\mathcal{E}$ if for every central face f of R, the cycle C^f that surrounds f in $\mathcal{E}_0$, is a two-sided cycle[3] in $\mathcal{E}$ that bounds a disc on one side, and the graph $R - V(C^f)$ is drawn on the other side. The following lemma is an easy corollary of Lemma 5.1:

Lemma 5.4 *There exists a ring $R \in \mathcal{R}$ that is embedded plainly in $\mathcal{E}$.*

Proof Suppose the contrary. For every ring $R \in \mathcal{R}$, let $C(R)$ be the cycle around a small face of R that violates the definition of a plainly embedded ring. Consider a graph $G' = G - \{v\} - M - S - \bigcup_{R \in \mathcal{R}} V(C(R))$. We shall prove that it has a large connected component D which contains $R' := R \setminus V(C(R))$ for each $R \in \mathcal{R}$. First, note that R' is itself connected and the boundary circles of R of are preserved

[3] Informally, a cycle C is *two-sided* in an embedding $\mathcal{E}$ if, while going along the cycle in the embedding and keeping track which incident face of the currently traversed edge is to the left and to the right, we end up in the same state as the one we started with. For a formal definition of one- and two-sided cycles in rotation systems, we refer to Chapter 4 of the book of Mohar and Thomassen [14].

in R'. Thus, it suffices to prove that for every two consecutive rings $R_1, R_2 \in \mathcal{R}$ (where R_1 is inside R_2, that is, the concentric cycles of W_3 that are part of R_1 have lower indices than the ones of R_2), the inner boundary circle of R_2 is connected to the outer boundary circle of R_1. Observe that the part of W_3 between these circles is a subdivision circular wall of circumference ℓ_3 and some positive height. Thus, there are $\ell_3 \geq k+1$ vertex-disjoint paths between the two boundary circles. These paths are disjoint with $\{v\}$, M, and $\bigcup_{R\in\mathcal{R}} V(C(R))$, and at least one of them must be disjoint with S. Hence, one of these paths is preserved in G'. We conclude that the claimed component D indeed exists.

We now aim at applying Lemma 5.1. Consider the surface where the embedding $\mathcal{E}$ embeds the graph $G - S$ and remove from this surface the images of $C(R)$ for all $R \in \mathcal{R}$. Let $D^{\mathcal{E}}$ be the component of this difference that contains the image of D. Since D is adjacent to every cycle $C(R)$, the images of $C(R)$ are in the closure of $D^{\mathcal{E}}$. By Lemma 5.1, as $|\mathcal{R}| > g$, there exists $R \in \mathcal{R}$ such that $C(R)$ bounds a disc that is disjoint with $D^{\mathcal{E}}$. In particular, the disk does not contain the image of $R' = R \setminus V(C(R))$. This is a contradiction with the choice of $C(R)$. □

Let $R \in \mathcal{R}$ be a plainly embedded ring, and let C_R be the central circle of R. A direct corollary of the definition of a plainly embedded ring is the following.

Corollary 5.5 *In the embedding $\mathcal{E}$, C_R is a two-sided cycle, and its incident edges of R are partitioned between the sides of C_R exactly as in the embedding $\mathcal{E}_0$.*

Proof Recall that the embedding $\mathcal{E}_0$, restricted to W_1, is the natural embedding of a subdivision of a wall as depicted in Fig. 5. Consequently, the ring R is embedded by $\mathcal{E}_0$ as in Fig. 8.

Traverse the cycle C_R in the embedding $\mathcal{E}_0$ and let e_1 and e_2 be two edges of $R \setminus E(C_R)$ that are incident to C_R and that are two consecutive edges on the same side of C_R in $\mathcal{E}_0$.

Furthermore, let P be the path between e_1 and e_2 in C_R that is not incident to any other edge of $R \setminus E(C_R)$ on the same side as e_1, and let f be the face of the embedding $\mathcal{E}_0$ restricted to R that is incident to e_1, e_2, and P; see also Fig. 10. Let e be the other edge of $R \setminus E(C_R)$ incident to P (the one that is drawn on the opposite side of C_R than e_1 in the embedding $\mathcal{E}_0$).

Since R is plainly embedded in $\mathcal{E}$, C^f bounds a disc in $\mathcal{E}$ that does not contain any edge from $R - V(C^f)$. This implies that C^f bounds a face in the embedding $\mathcal{E}$ restricted to R. Thus, as we traverse P in $\mathcal{E}$, the edges e_1 and e_2 are on one side, and e is on the other side. Since the choice of e_1, e_2, and f is arbitrary, the claim follows. □

Corollary 5.5 allows us to speak about the *inner* and *outer* side of C_R in $\mathcal{E}$: the sides of C_R that contain incident edges inside C_R and outside C_R in $\mathcal{E}_0$.

A *bridge* is a connected component C of $G - V(R)$, together with the edges joining C with $V(R)$. Furthermore, an edge $e \notin E(R)$ with both endpoints in $V(R)$ is also a bridge on its own. For a bridge B, the vertices of $V(B) \cap V(R)$ are *attachment points*.

A bridge B is *central* if it has at least one attachment point, but all its attachment points lie in $V(C_R)$. Note that, since R is contained in W_2, a central bridge is disjoint

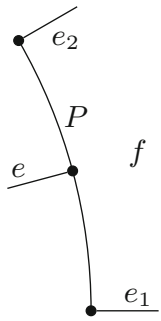

Fig. 10 Illustration for the proof of Corollary 5.5

with M and in the embedding $\mathcal{E}_0$ it is drawn inside one of the small faces of R incident to C_R. In particular, a central bridge is a subgraph of the territory of R.

Let H be the subgraph of $G - M$ that consists of: the part of $G - M$ that is enclosed on the same side of C_R in the embedding $\mathcal{E}_0$ as the vertex v (i.e., on the disc, flat side of C_R), together with all central bridges. In other words, H is a subgraph of $G - M$ induced by the vertices of C_R and all connected components of $G - M - V(C_R)$ that are contained in the same connected component of $G - M$ as C_R, except for the connected component of $G - M - V(C_R)$ that contains the outermost concentric cycle of R. The boundary of H, denoted ∂H, is the set of vertices of H that have incident edges of G that do not belong to H.

Let us now make a few important observations about the graph H. First, by the definition of H (in particular, inclusion of all central bridges) and the choice of W_2 (so that G_2 is disjoint from any neighbors of M), all boundary vertices of H lie on C_R, which induces a two-sided curve both in $\mathcal{E}_0$ and $\mathcal{E}$. Second, since the embedding of $\mathcal{E}_0$ restricted to W_1 is the natural embedding of a wall (as in Fig. 5), the embedding $\mathcal{E}_0$ restricted to H is a planar embedding $\mathcal{E}_H$ of H that keeps all vertices of ∂H on the infinite face. Furthermore, there exists a closed simple curve γ_0 in $\mathcal{E}_H$ that visits all vertices of ∂H in the same cyclic order as they appear on C_R and, apart from these vertices, is contained in the infinite face of this planar embedding.

Armed with these observations, we now modify the embedding $\mathcal{E}$ of $G - S - \{v\}$ as follows. We start by deleting all edges of H and vertices of $V(H) \setminus \partial H$. Let $\mathcal{E}^1$ be the resulting embedding. We claim that there exists a face f in $\mathcal{E}^1$ that contains all the vertices of ∂H and, furthermore, there exists a closed curve γ in the embedding $\mathcal{E}$ that visits all vertices of ∂H exactly in the order how they appear on C_R and, apart from these vertices, is contained in f. To see this, recall that R is drawn plainly in the embedding $\mathcal{E}$ and, consequently, the area on the inner side of the cycle C_R in the embedding $\mathcal{E}$ (i.e., a sufficiently narrow strip on the inner side of C_R) belongs to a single face of $\mathcal{E}^1$. This face of $\mathcal{E}^1$, denoted f, contains the image of C_R in its closure in the embedding $\mathcal{E}$. The image of C_R forms the curve γ as specified above.

We now merge the embeddings $\mathcal{E}^1$ and $\mathcal{E}_H$ into a single embedding along the curves γ_0 and γ. That is, we start with $\mathcal{E}^1$, then we embed H according to the embedding $\mathcal{E}_H$ into the face f, and finally we identify the corresponding vertices of ∂H using the existence of the curves γ and γ_0. In this manner, we obtain an embedding $\mathcal{E}^2$ of

$G - (S \setminus V(H))$ into the same surface as the embedding $\mathcal{E}$, concluding the proof that $v \in V(H)$ is irrelevant.

This completes the proof of Theorem 1.3.

6 Conclusions

In this work, we have developed fixed-parameter algorithms for the Genus Vertex Deletion problem with solution size and treewidth parameterizations, putting particular effort into optimizing the dependency on the treewidth in the running time bound.

We remark that, although our formal statement of Genus Vertex Deletion involves only bounding the Euler genus of the output graph, only minor changes to our algorithms are required if one demands the final graph to be embeddable in an *orientable* surface of some genus. In terms of combinatorial embeddings studied in Sect. 3, an embedding is orientable if the set of flags can be partitioned into two parts such that every orbit of σ, θ, and ϕ contains two flags from different sets. The crucial observation is that deleting an edge, drawing an edge along a face boundary, and suppressing a size-4 vertex that is not isolated, applied to an orientable embedding results in an embedding that is also orientable. Consequently, if we allow only orientable embeddings in the dynamic programming algorithm of Sect. 4, we obtain the desired variant of Theorem 1.2 for orientable surfaces. Finally, the arguments of Sect. 5 operate in the language of modifying an embedding in a fixed surface; therefore, also without any changes, they yield a variant of Theorem 1.3 for orientable surfaces.

We further note that the dynamic programming behind Theorem 1.2 can be trivially extended to the weighted setting, where deleting a vertex v incurs a penalty $w(v) \in \mathbb{R}_+$. On the other hand, Theorem 1.3 can only be extended to support integer weights if k is defined as the total budget for vertex deletions.

We would like to conclude with two open questions stemming from our research.

First: Can we obtain a $2^{\mathcal{O}(C_g k \log k)} n$-time algorithm for Genus Vertex Deletion, following the ideas of [7] for the planar case? Our bounded treewidth routine suits such an algorithm, but the irrelevant vertex argument does not.

Second, and more challenging: What can we say about a possible dependency on the parameter k for the problem of deleting k vertices to an arbitrary minor-closed graph family? A similar question can be asked for the parameter treewidth. Here, the main challenge is that it is harder to certify being H-minor-free for an arbitrary graph H, while one can certify being of bounded genus by giving a corresponding embedding.

Acknowledgements We thank an anonymous reviewer for their very careful review and multiple comments that helped improve the presentation.

References

1. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**(6), 1305–1317 (1996). https://doi.org/10.1137/S0097539793251219
2. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015). https://doi.org/10.1007/978-3-319-21275-3
3. Demaine, E.D., Hajiaghayi, M.T., Kawarabayashi, K.: Algorithmic graph minor theory: decomposition, approximation, and coloring. In: 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005, pp. 637–646. IEEE Computer Society (2005). https://doi.org/10.1109/SFCS.2005.14
4. Demaine, E.D., Hajiaghayi, M.: Linearity of grid minors in treewidth with applications through bidimensionality. Combinatorica **28**(1), 19–36 (2008). https://doi.org/10.1007/s00493-008-2140-4
5. Diestel, R.: Graph Theory, volume 173 of Graduate Texts in Mathematics, 5th edn. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-53622-3
6. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. **63**(4), 512–530 (2001). https://doi.org/10.1006/jcss.2001.1774
7. Jansen, B.M.P., Lokshtanov, D., Saurabh, S.: A near-optimal planarization algorithm. In: Chekuri, C. (ed.) 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, pp. 1802–1811. SIAM (2014). https://doi.org/10.1137/1.9781611973402.130
8. Kawarabayashi, K.: Planarity allowing few error vertices in linear time. In: 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, pp. 639–648. IEEE Computer Society (2009). https://doi.org/10.1109/FOCS.2009.45
9. Kawarabayashi, K., Mohar, B., Reed, B.A.: A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, pp. 771–780. IEEE Computer Society (2008). https://doi.org/10.1109/FOCS.2008.53
10. Kloks, T.: Treewidth, Computations and Approximations, volume 842 of LNCS. Springer, Berlin (1994). https://doi.org/10.1007/BFb0045375
11. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. **20**(2), 219–230 (1980). https://doi.org/10.1016/0022-0000(80)90060-4
12. Marx, D., Schlotter, I.: Obtaining a planar graph by vertex deletion. Algorithmica **62**(3–4), 807–822 (2012). https://doi.org/10.1007/s00453-010-9484-z
13. Mohar, B.: A linear time algorithm for embedding graphs in an arbitrary surface. SIAM J. Discrete Math. **12**(1), 6–26 (1999). https://doi.org/10.1137/S089548019529248X
14. Mohar, B., Thomassen, C.: Graphs on Surfaces. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press (2001). https://jhupbooks.press.jhu.edu/title/graphs-surfaces. Accessed 03 June 2019
15. Pilipczuk, M.: A tight lower bound for vertex planarization on graphs of bounded treewidth. Discrete Appl. Math. **231**, 211–216 (2017). https://doi.org/10.1016/j.dam.2016.05.019
16. Stahl, S., Beineke, L.W.: Blocks and the nonorientable genus of graphs. J. Graph Theory **1**(1), 75–78 (1977). https://doi.org/10.1002/jgt.3190010114