

Embeddings of Graphs of Fixed Treewidth and Bounded Degree*

Jonathan L. Gross

Columbia University, Department of Computer Science NY 10027 USA, New York, USA

Received xxx 2012, accepted xxx 2013, published online xxx 2013

Abstract

Let \mathcal{F} be any family of graphs of fixed treewidth and bounded degree. We construct a quadratic-time algorithm for calculating the genus distribution of the graphs in \mathcal{F} . Within a post-order traversal of the decomposition tree, the algorithm involves a full-powered upgrading of *production rules* and *root-popping*. This algorithm for calculating genus distributions in quadratic time complements an algorithm of Kawarabayashi, Mohar, and Reed for calculating the minimum genus of a graph of bounded treewidth in linear time.

Keywords: genus distribution, partial genus distribution, treewidth, tree decomposition

Math. Subj. Class.: 05C10

Version: 13:34 August 22, 2013

*This paper was presented at the AMS Meeting of January, 2012 in Boston, MA.
E-mail address: gross@cs.columbia.edu (Jonathan L. Gross)

1 Introduction

For $i = 0, 1, 2, \dots$, let $g_i(G)$ be the number of topologically distinct cellular embeddings of the graph G in the orientable surface S_i of genus i . The **genus distribution** of the graph G is the sequence of numbers

$$g_i(G) : i = 0, 1, \dots$$

The smallest and largest numbers i such that $g_i(G)$ is positive are called the **minimum genus** and the **maximum genus**, respectively, of the graph G . It is easily proved and well-known that there are cellular embeddings of a graph G in every surface whose genus lies between the minimum and the maximum. The set of numbers between (and including) the minimum and maximum genus is called the **genus range** of G .

The main objective of this paper is to derive a quadratic-time algorithm to calculate the entire genus distribution for any family of simple graphs with fixed treewidth and bounded degree. Since the second barycentric subdivision of a general graph is a simple graph with the same genus distribution as the general graph to which it is homeomorphic, it follows that this can be extended to general graphs by subdividing edges.

This paper also introduces a general form of partial genus distribution for arbitrarily large degree and for arbitrary root-subgraphs, i.e., beyond vertices and edges, a relativized form of partial genus distribution modulo a fixed rotation system for its root subgraph, and a general form of production rules for iterative reassembly of a given graph from one or more small subgraphs.

BASIC RESULTS ON GENUS DISTRIBUTION

Five fundamental papers [10, 7, 18, 15, 20] of the present author and his co-authors Khan and Poshni have established methods for calculating the genus distribution of a graph that is constructed by various kinds of amalgamation of two graphs of known genus distribution. These papers also establish ways to calculate the genus distributions of chains and cycles of copies of graphs of known genus distribution. The methods developed in these papers include *recombinant strands*, *partitioned genus distributions*, and *production rules*.

More recently, combining these calculation methods with the algorithmic techniques of *post-order traversal* and *root-popping* has facilitated the calculation of genus distributions for 3-regular outerplanar graphs [8], for 4-regular outerplanar graphs [19], and for 3-regular Halin graphs [9]. Combining these same calculation methods with edge-addition [16] has led to the genus distribution of mesh graphs of the form $P_3 \times P_n$.

CONNECTIONS OF TREEWIDTH TO EMBEDDING PROBLEM

Since the introduction of the concept of *treewidth* by Robertson and Seymour, bounding the treewidth has been widely used to obtain polynomial-time algorithms for problems that are otherwise NP-hard. In particular, deciding whether an arbitrarily selected graph can be embedded in a given surface is NP-complete [25]; however, for any class of graphs

of bounded treewidth, Kawarabayashi, Mohar, and Reed [14] have derived a linear-time algorithm for calculating the minimum genus.

Although outerplanar graphs have treewidth 2, and although Halin graphs and $P_3 \times P_n$ meshes have treewidth 3 (see [3]), treewidth plays no explicit role in the calculation of genus distributions in any of the papers just mentioned. Indeed, the five fundamental papers on graph amalgamations cited above include applications to graphs of arbitrarily high treewidth and arbitrarily high degree. Nonetheless, the pastings in those papers occur in localities of the amalgamand graphs in which the treewidth and degree are bounded.

In the present paper, various key ideas from the earlier papers are abstracted, generalized, and combined with treewidth to yield a quadratic-time algorithm for the genus distribution of the graphs in any family of graphs of bounded degree and bounded treewidth. It will be apparent that as treewidth and degree increase, the multiplicative constant of the quadratic term grows rapidly. Accordingly, one anticipates continued interest in the derivation of special methods for calculating the genus distributions of graph families of special interest.

TERMINOLOGY

In what follows, a **graph** is taken to be connected and simple, unless something else can be inferred from the immediate context. We use V_G and E_G to denote the vertex set and edge set of a graph G .

The embeddings are in oriented surfaces. Terminology used here is predominantly consistent with [13] and [1]. See also [17] and [28]. We abbreviate “face-boundary walk” as **fb-walk**.

DEF. In this paper, a **subgraph-rooted** graph is a triple (G, H, u) , where H is a subgraph of G and $u \in V_H$. The third parameter u is called the **pivot**. Sometimes, the form (G, H) with no pivot is used. If $H \cong K_1$ or $H \cong K_2$, then the graph is **vertex-rooted** or **edge-rooted**, respectively. When the context clarifies the meaning, the graph may simply be called **rooted**.

REMARK. Pivot vertices are used here to change the root subgraph as a sequence of graphs is formed in the process of reassembly of a given graph. In [8], we achieved a change of root with what we called “root-splitting”. In [9], a change of root was accomplished by “pie-merges”.

OUTLINE OF THIS PAPER

Section 2 of this paper describes treewidth from a perspective that is relevant to its use in the algorithm. Section 3 describes how a decomposition tree is used to analyze a graph into *fragments* to be amalgamated. Section 4 introduces a highly general way of partitioning the genus distribution of a graph; it shows that the number of cells of the partition depends only on the treewidth and the maximum degree, and not on the number of vertices of that graph. Section 5 shows that the number of embeddings of an amalgamated

graph depends on the degrees of the vertices of the subgraph of amalgamation, and not on the number of vertices of the graph. Section 6 describes production rules, as they occur in the algorithm. Section 7 describes and analyzes the genus distribution algorithm. Section 8 offers some conclusions about the algorithm and opportunities for future research.

This paper is largely self-contained, except for some details of the well-established methods of constructing productions (which is quite necessary for the algorithm), as in [10] and [18]. Prior experience with calculating genus distributions of graph amalgamations, especially as in [8] and [9], is likely to be quite helpful.

2 Treewidth

The usual definition of *treewidth* is based on the concept of *tree decomposition*. These are both due to Robertson and Seymour [21]. An excellent exposition is given by [3]. For applications of treewidth to topological graph theory, see [17].

DEF. Let $G = (V, E)$ be a graph, and T a tree with nodes $1, 2, \dots, s$. Let $\mathcal{X} = \{X_i \mid 1 \leq i \leq s\}$ be a family of subsets of V (associated with the respective nodes $1, 2, \dots, s$) whose union is V such that

- the induced graph on the set of images in T of each vertex of V is a subtree of T ;
- for every edge uv in the graph G , there is a node i in the tree T such that both u and v are members of X_i .

Then the pair (\mathcal{X}, T) is a **tree decomposition** of G , and the tree T is called a **decomposition tree** for G .

TERMINOLOGY. For the sake of clarity, we will refer to “vertices” and “edges” in the graph G and to “nodes” and “lines” in the tree T .

ABUSE OF NOTATION. Throughout this paper, we refer to the sets X_i of a decomposition tree as “nodes”.

DEF. The **width** of a tree decomposition (\mathcal{X}, T) equals

$$\max \left\{ |X_i| \mid 1 \leq i \leq |V_T| \right\} - 1$$

DEF. The **treewidth** of a graph G is the smallest k such that G has a tree decomposition of width k .

Proposition 2.1. *A connected graph has treewidth 1 if and only if it is a tree.* □

Proposition 2.2 ([27]). *A connected graph has treewidth 2 if and only if it contains a cycle and does not contain a K_4 -minor.* □

Proposition 2.3. *Every graph of treewidth 2 is planar.*

Proof. A non-planar graph has either K_5 or $K_{3,3}$ as a minor. Both these Kuratowski graphs have K_4 as a minor. By Proposition 2.2, a graph with a K_4 -minor cannot have treewidth 2. \square

TREewidth CHARACTERIZATION WITH k -TREES

An alternative characterization of treewidth, in terms of k -trees (see, for instance, [4]), is the starting point of our present approach to genus distributions:

DEF. A k -tree is defined recursively:

- The complete graph K_{k+1} is a k -tree.
- If G is a k -tree and C is a k -clique in G , then the graph obtained by joining a new vertex to the vertices of C is a k -tree.

Proposition 2.4. *The treewidth of a graph G is the least number k such that G is a subgraph of a k -tree.*

Proof. The proof is a direct consequence of the definitions. \square

FULL DECOMPOSITION TREES

DEF. A **full decomposition tree of width k** for a graph G is a decomposition tree T in which

- every node has $k + 1$ vertices, and
- every pair of adjacent nodes intersects in k vertices.

We observe that each line of a full decomposition tree corresponds to the k vertices shared by the two nodes whose adjacency is represented by that line.

Proposition 2.5. *Let T be a full decomposition tree of treewidth k for an n -vertex graph G . Then $|V_T| = n - k$.*

Proof. Each node of the decomposition tree T contains $k + 1$ vertices of G . Each line of T corresponds to k vertices of G . Since V_G is the union of the nodes of T , it follows that

$$\begin{aligned} n = |V_G| &= |V_T|(k + 1) - |E_T|k \\ &= |V_T|(k + 1) - (|V_T| - 1)k \\ &= |V_T| + k \\ \therefore |V_T| &= n - k \end{aligned} \quad \square$$

Running Example – Part 1. Figure 2.1 shows an 8-vertex graph of treewidth 2 and a full decomposition tree of width 2. We observe that the number of nodes of the full decomposition tree is $8 - 2 = 6$.

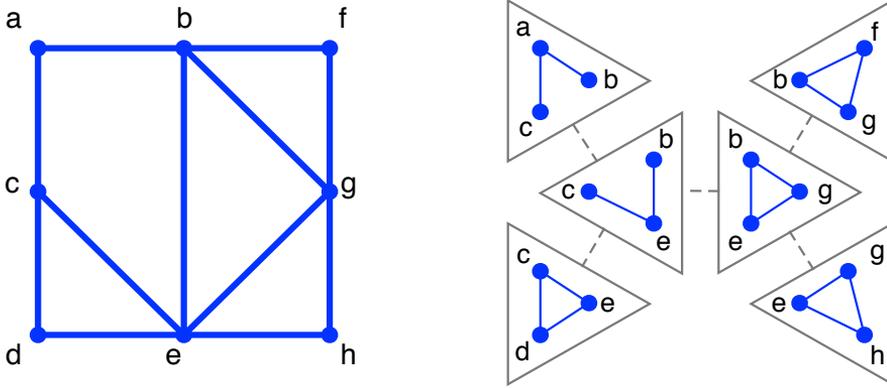


Figure 2.1: A graph, and a full decomposition tree of width 2, represented as a subgraph of a 2-tree.

The figure on the left is the graph itself. Each gray triangle on the right represents a 3-clique of the 2-tree. Each dashed gray line represents an adjacency of two 3-cliques in the 2-tree. Each node of the full decomposition tree is the set of vertices lying within one of the gray triangles. The edges drawn within the gray triangles are a reminder of the adjacencies in the graph itself. In §3, the subgraphs shown within the gray triangles are called *node-fragments*.

Theorem 2.6. *Let G be a graph of treewidth k . Then G has a full decomposition tree of width k .*

Proof. Let T be a decomposition tree of width k for a graph G of treewidth k . If the vertex set from G in one of two adjacent nodes of T is contained in the other, then contract the line of T that joins those two nodes, and eliminate the smaller node. We observe that this operation does not change the width of the resulting tree. Iterate this operation until the resulting decomposition tree for G has the following property:

- (P_1) For every node $X_i \in V_T$ and for each of its neighboring nodes X_j , there is a vertex in X_i that does not lie in the node X_j .

For simplicity, we assume that the initial tree T already has property P_1 . Since the maximum size of a node of T is $k + 1$, and since no two adjacent nodes are identical, it follows that the intersection of any pair of adjacent nodes contains at most k vertices.

If some node X_i of tree T contains fewer than $k + 1$ vertices, then choose a vertex from any neighboring node and insert a copy of it into node X_i . If the node X_i now contains every vertex in that neighbor, then contract the line of T that joins those two nodes. Iterate until every node of the resulting tree has $k + 1$ vertices of G . By construction, this tree has property P_1 . We may now assume that the initial tree T also has this property:

- (P_2) Every node $X_i \in V(T)$ has $k + 1$ vertices of G .

REMARK. It would be possible to design an algorithm for genus distribution that does not require the given decomposition tree to be converted into a full decomposition tree T . The reason we perform this conversion here is to simplify subsequent discussion, for instance, of the use of *partial genus distributions* in §4.

Now suppose that X_i and X_j are adjacent nodes of T such that

$$\begin{aligned} X_i &= \{v_1, \dots, v_m, u_{m+1}, \dots, u_{k+1}\} \\ X_j &= \{v_1, \dots, v_m, w_{m+1}, \dots, w_{k+1}\} \quad \text{and} \\ X_i \cap X_j &= \{v_1, \dots, v_m\} \quad \text{where } m < k. \end{aligned}$$

Then replace the line (X_i, X_j) in T by the node path

$$\begin{aligned} X_i &= \{v_1, \dots, v_m, u_{m+1}, \dots, u_{k+1}\} \\ X_i^{(1)} &= \{v_1, \dots, v_m, w_{m+1}, u_{m+2}, \dots, u_{k+1}\} \\ X_i^{(2)} &= \{v_1, \dots, v_m, w_{m+1}, w_{m+2}, u_{m+3}, \dots, u_{k+1}\} \\ &\quad \dots \\ X_i^{(k-m)} &= \{v_1, \dots, v_m, w_{m+1}, \dots, w_k, u_{k+1}\} \\ X_j &= \{v_1, \dots, v_m, w_{m+1}, \dots, w_{k+1}\} \end{aligned}$$

The resulting tree is a full decomposition tree for G of width k . □

Corollary 2.7. *For any fixed treewidth k , there is an algorithm to construct a full decomposition tree for a given graph G of treewidth k in linear time in $|V_G|$.*

Proof. A linear-time algorithm to construct a decomposition tree T of width k for the graph G is given by [2]. Since the number of edges of such a tree is linear in $|V_G|$, it follows that a full decomposition tree for G of width k can be constructed from T within linear time. □

Corollary 2.8. *Let G be a graph of treewidth k , and let T be a full decomposition tree for G . Then there is a k -tree T^k with the following property:*

Each node of T is a $(k + 1)$ -clique of the k -tree T^k .

Proof. This is implied by the definition of a full decomposition tree. □

The genus distribution of a graph G is to be derived from the partial genus distributions of the induced graphs (in G) on the vertices in the respective nodes of the tree T , by iterative amalgamation. Various key ideas for deriving the genus distribution of a graph G by iterative amalgamation of a set of subgraphs of G are developed in [6], [7], [8], [9], [10], [15], [18], [19], and [20].

3 Fragments and Amalgamations

The algorithm to calculate the genus distribution of a graph G with bounded treewidth and bounded degree reassembles the graph G by iteratively amalgamating induced subgraphs on the nodes of a full decomposition tree for G .

DEF. A **node-fragment of a graph** G with respect to a decomposition tree T is the induced subgraph in G on the set of vertices of G that lie within a single node of T . For a leaf-node of T , the corresponding node-fragment may be called a **leaf-fragment**.

DEF. Let (G, H, u) and (G', H', u') be an ordered pair of disjoint subgraph-rooted graphs, and let $\eta : H - u \rightarrow H' - u'$ be a graph isomorphism. **Graph amalgamation** is the operation that forms a new graph $G *_\eta G'$ from $G \cup G'$ by merging the subgraphs $H - u$ and $H' - u'$ as prescribed by η . In this paper, the subgraph H' becomes the new root subgraph, and a new pivot is chosen according to the post-order of the decomposition tree. This is illustrated in Part (2) of the Running Example.

NOTATIONAL CONVENTION. We denote the vertices and edges of the subgraph $H *_\eta H'$ of the amalgamated graph $G *_\eta G'$ by the same names as in the subgraph H of graph G , the first amalgamand. The vertices and edges that are contributed by only one amalgamand retain the names used in that amalgamand.

DEF. A **fragment of a graph** G with respect to a decomposition tree T for G is either a node-fragment or a **compound fragment**, by which we mean the result of amalgamating any two fragments across the induced subgraphs of the vertices that lie in both of two adjacent nodes of the tree T . Every amalgamation in the reassembly of G from its fragments corresponds to a line of the decomposition tree T .

Given a graph G and a full decomposition tree T of width k , with T envisioned as drawn in the plane, it is clear that G can be reassembled by iteratively amalgamating fragments on pairs of vertices.

- We fix an arbitrary leaf-node of the tree T as a root of T , and we determine a post-order traversal of T based at that root-node.
- We observe that during a post-order traversal, every line of T is traversed twice. We amalgamate across a line whenever the second traversal of that line occurs.
- Each fragment of the graph G is a rooted subgraph of G . In any fragment F , the root subgraph is the induced graph in G on the node of T in which the fragment F meets the fragment to which it will be amalgamated. In a non-leaf-fragment, the root-subgraph is to be chosen according to the post-order of T .

LABELING FRAGMENTS AND SELECTING ROOTS OF FRAGMENTS

In order to discuss the process of iterative amalgamation, it is helpful to have some rules for assigning labels to fragments.

DEF. The **label of a fragment** F for a graph G with a full decomposition tree T of width k is of the form $G[x_1, \dots, x_q; r_1, \dots, r_k; r_{k+1}]$, where the vertex set of the fragment F is

$$\{x_1, \dots, x_q, r_1, \dots, r_k, r_{k+1}\}$$

- The vertices r_1, \dots, r_k, r_{k+1} are from whatever node of the fragment F will be pasted to another fragment in the next amalgamation involving F that occurs in the post-order.
- The root-subgraph of fragment F is the induced subgraph on vertices r_1, \dots, r_k, r_{k+1} .
- Vertex r_{k+1} is the pivot.
- The vertices x_1, \dots, x_q are the remaining vertices of fragment F .

DEF. In the course of reassembly of a graph from its node-fragments by iterative amalgamation, for each fragment F , whether a node-fragment or a compound fragment, its **partner fragment** F' is the fragment to which F is amalgamated during the reassembly. Similarly, each node-fragment M has a **partner node-fragment** M' .

Running Example – Part 2. The root subgraph of a node-fragment is the node-fragment itself. In each node-fragment of Figure 3.1, the two vertices with bold labels are the first two to be pasted to another node. The third vertex in the node is the initial pivot. These are determined by the post-order traversal. We have taken the node $G[; c, e; d]$ at the lower left of the tree as the root-node of the decomposition tree and used a counter-clockwise traversal.

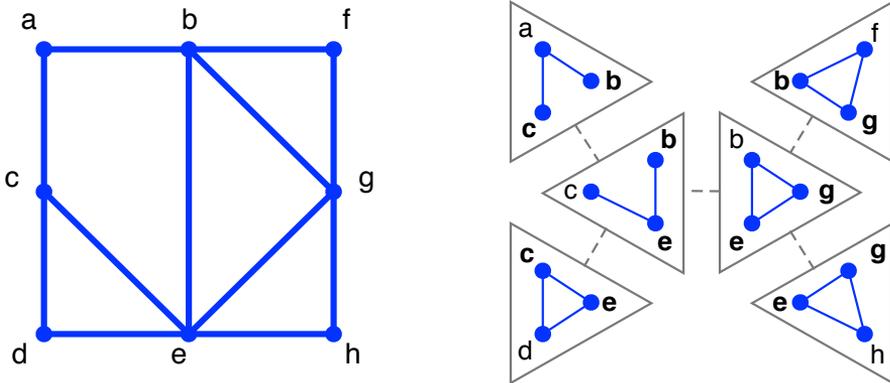


Figure 3.1: Roots of a decomposition 2-tree for a graph G .

Here is the reassembly sequence for the graph of Figure 3.1.

1. The first node-fragment is $F_0 = G[; e, g; h]$.
2. The node-fragment (at the center-right) adjacent to F_0 is labeled $G[; e, g; b]$. When node-fragments $F_0 = G[; e, g; h]$ and its partner $F'_0 = G[; e, g; b]$ are amalgamated, the resulting fragment is

$$F_1 = G[h; b, g; e]$$

3. When fragment $F_1 = G[h; b, g; e]$ is amalgamated with node-fragment $F'_1 = G[; b, g; f]$, the resulting fragment is

$$F_2 = G[f, h; b, e; g]$$

4. When fragment $F_2 = G[f, h; b, e; g]$ is amalgamated with node-fragment $F'_2 = G[; b, e; c]$, the resulting fragment is

$$F_3 = G[f, g, h; b, c; e]$$

5. When fragment $F_3 = G[f, g, h; b, c; e]$ is amalgamated with node-fragment $F'_3 = G[; b, c; a]$, the resulting fragment is

$$F_4 = G[a, f, g, h; c, e; b]$$

6. When fragment $F_4 = G[a, f, g, h; c, e; b]$ is amalgamated with node-fragment $F'_4 = G[; e, c; d]$, the resulting fragment is

$$F_5 = G[a, b, f, g, h; c, e; d] = G$$

We observe the following properties of each of the amalgamation in the sequence:

- The pivot of each of the amalgamand fragments F_i and F'_i is not a vertex of the other amalgamand fragment.
- The root-subgraph of the amalgamated fragment F_{i+1} is the root-subgraph of one of the amalgamand fragments F_i and F'_i . More precisely, it is whichever of the two root-subgraphs will be used whenever F_{i+1} is amalgamated to F'_{i+1} .
- The pivot of the amalgamated fragment F_{i+1} is the unique vertex in the root-subgraph that will *not* be merged with another vertex in the next amalgamation step involving fragment F_{i+1} .

REMARK. Although the partner fragments F'_i were all node-fragments in this example, this would not be the case with a more complicated decomposition tree.

4 Partitioning the Genus Distribution

Partitioning the embeddings of a rooted graph has proven to be a highly useful technique in calculating genus distributions. A surface-by-surface inventory of the cells of the partition is called a *partitioned genus distribution*. The criteria for partitioning has been the incidence of fb-walks on the roots.

In the simplest case, we have a graph (G, v) rooted at a single 2-valent vertex v . We can then partition the genus distribution sequence $\{g_i(G) : i = 0, 1, \dots\}$ into two **partial genus distribution sequences**

$$d_i(G, v) : i = 0, 1, \dots \quad \text{and} \quad s_i(G, v) : i = 0, 1, \dots$$

where

- $d_i(G, v)$ is the number of embeddings of G in which two distinct fb-walks are incident on the root-vertex v ; and
- $s_i(G, v)$ is the number of embeddings of G in which a single fb-walk is twice incident on the root-vertex v .

The prototypical approach to calculating these distributions is to construct simultaneous recursions for d_i and s_i and to obtain g_i by adding their solutions.

In the next smallest case, that of two 2-valent roots u and v , there are ten different partial genus distributions. For instance, in [10] and [7], we have defined

- $dd'_i(G, u, v)$ as the number of embeddings of G in S_i in which there are two different fb-walks incident on u , one of which is also incident on v , and another fb-walk incident on v that is not incident on u .
- $sd'_i(G, u, v)$ as the number of embeddings of G in S_i in which one fb-walk is twice incident on u and also incident on v , and another fb-walk is incident on v and not on u .

When there are multiple roots and/or a root-subgraph, the number of partial genus distributions sequences grows rapidly with the total number of root-vertices or of vertices in the root-subgraph. We can partition the genus distribution of a graph G into as many partial genus distributions as needed, the sum of which is the genus distribution sequence $\{g_i(G) : i = 0, 1, \dots\}$.

In the general case now under consideration, the genus distribution of a subgraph-rooted graph (G, H, u) is partitioned here according to the cyclic sequence of incidences of fb-walks on the vertices of the root-subgraph H and on the pivot u . This section provides the details, an example, and an upper bound on the number of restricted sequences in the partition corresponding to a graph of treewidth k and maximum degree Δ . This upper bound depends only on the treewidth and maximum degree of a graph, and not on its number of vertices.

DEF. A ***gapped word*** on the root-vertices r_1, \dots, r_k, r_{k+1} of a rooted graph (G, H, r_{k+1}) is a word on the alphabet $\{r_1, \dots, r_k, r_{k+1}, \bullet\}$ that contains no two consecutive occurrences of the bullet \bullet , which is called a ***gap***. Each gapped word represents the cyclic order in which the various root-vertices occur in an fb-walk of an embedding. When two root-vertices r_i and r_j are separated by a gap symbol, it means that the subwalk of the corresponding fb-walk between r_i and r_j contains one or more non-root-vertices. When two root-vertices r_i and r_j are adjacent, it means that there is an edge $r_i r_j$ traversed by the fb-walk.

Two gapped words are equivalent if one is a cyclic permutation of the other. The principal representative of each equivalence class of gapped words is the one that is lexicographically first. We regard the bullet as lexicographically last, i.e., after all the root-vertices.

DEF. A multi-set of principal gapped words (written as a tuple) is called a ***root-phrase*** if

- each root-vertex occurs in the union of the gapped words as many times as its valence in G ;

- the principal gapped words are in the order of non-increasing size, with lexicographic order used for tie-breaking.

Each root-phrase represents a cell of the partition of the embeddings of (G, H, u) . Moreover, each root-phrase may be subscripted by an integer that represents the genus of a surface.

These two examples illustrate how a root-phrase is used in our generalization of partial genus distributions. (With larger root-subgraphs, some components of a root-phrase may have integer coefficients.)

Example 4.1. For non-adjacent roots u and v , $dd'_i(G, u, v)$ is represented by the root-phrase $(u\bullet v\bullet, u\bullet, v\bullet)_i$.

Example 4.2. For non-adjacent roots u and v , $sd'_i(G, u, v)$ is represented by the root-phrase $(u\bullet u\bullet v\bullet, v\bullet)_i$.

DEF. The *partitioned genus distribution* of (G, H, u) is a linear combination of the subscripted root-phrases, in which the coefficient of each subscripted root-phrase is the number of oriented embeddings of G corresponding to that root-phrase, in the surface whose genus equals the particular subscript.

DEF. We use the abbreviation *pgd* for partitioned genus distribution.

DEF. We observe that the restriction of the pgd of (G, H, u) to a single root-phrase, taken over all subscripts in the genus range, is the inventory of embeddings within a single partition cell, i.e., the cell corresponding to the given root-phrase. This inventory is called a *partial genus distribution* of (G, H, u) . The word *partial*, when used here as a noun, is a synonym for root-phrase. In this sense, we may say that the pgd is a linear combination of the subscripted partials.

Running Example – Part 3. The node-fragment $G[; e, g; h]$ has the following pgd:

$$(egh, ehg)_0 \tag{4.1}$$

The only embedding of that fragment has two (oriented) fb-walks, egh and ehg . Since all the vertices in the only two fb-walks are root-vertices, and since the root-vertices are mutually adjacent, there are no gaps in the gapped words of the only root-phrase. Similarly the node-fragment $G[; e, g; b]$ has the following pgd:

$$(beg, bge)_0 \tag{4.2}$$

The compound fragment $G[h; b, g; e]$ has the pgd

$$(be\bullet g, bge, eg\bullet) + (bg\bullet e, beg, e\bullet g)_0 + (beg\bullet ebge\bullet g)_1 + (be\bullet gebg\bullet eg)_1 \tag{4.3}$$

To see this most easily, one draws the four embeddings and traces the fb-walks in each embedding.

NOTATION. We use $p(n)$ to denote the number of partitions of a positive integer n . We recall the asymptotic formula of Hardy and Ramanujan:

$$p(n) \sim \frac{1}{4n\sqrt{3}} e^{\pi\sqrt{\frac{2n}{3}}} \quad \text{as } n \rightarrow \infty$$

We use the number of partitions toward an upper bound on the number of partials of a rooted graph.

NOTATION. We denote the **degree of a vertex** v of a graph Y by $\delta_Y(v)$. If Y is a subgraph of a graph X , this convention distinguishes the degree of vertex v in the subgraph Y from its degree in X .

Theorem 4.1. *Let \mathcal{F} be a family of graphs of treewidth k and maximum degree Δ , each of which is rooted on an isomorphic copy of a fixed graph H , in which $V_H = \{r_1, \dots, r_k\}$. Then the number of partials associated with amalgamating two arbitrary members of \mathcal{F} across root-subgraph H is at most*

$$\frac{[k\Delta]!}{(\Delta!)^k} \cdot p(k\Delta) \cdot 2^{k\Delta}$$

That is, it has an upper bound that is independent of the number of vertices of the graphs being amalgamated.

Proof. Here we regard a root-phrase as a “raw” character string in the alphabet $V_H = \{r_1, \dots, r_k\}$, into which gaps and commas are eventually inserted. The length of a raw character string for a graph G is $\sum_{i=1}^k \delta_G(r_i)$. (We need to use δ_G rather than δ_H because we are concerned with the degree of each vertex of H within the graph G , not just within H .) Thus, the cardinality of the set of raw character strings is

$$\frac{[\sum_{i=1}^k \delta_G(r_i)]!}{\prod_{i=1}^k \delta_G(r_i)!} \leq \frac{[k\Delta]!}{(\Delta!)^k}$$

We may insert the commas into each raw character string so that the number of characters between two successive commas is non-increasing, as one reads the string from beginning to end. The cardinality of the set of comma-enriched raw character strings so obtained is at most $p(k\Delta)$. The number of ways to insert from 0 to $k\Delta$ gaps into such a comma-enriched character string so that

- no two gaps are adjacent,
- no gap occurs immediately after a comma, and
- no gap occurs at the beginning of the first word

is at most $2^{k\Delta}$. We have pre-normalized the root-phrases so that no gap occurs at the beginning of a word and so that the gapped words are in the order of non-increasing length. Thus, every normalized gapped word is a member of the set of such gap-enriched, comma-enriched character strings. This establishes the upper bound of the theorem. \square

When we are reassembling a graph G of treewidth k and maximum degree Δ from the node-fragments of a full decomposition tree, the k -vertex root-subgraph across which two fragments are amalgamated varies by one vertex at a time, as we traverse the decomposition tree. The isomorphism type of the root-subgraph across which we merge fragments may vary from one amalgamation to the next. Thus, rather than only with the partials associated with a fixed root-subgraph H , we need to be concerned with the set of all partials for root-subgraphs with a given number of vertices.

Theorem 4.2. *Let G be a graph of treewidth k and maximum degree Δ . Then the number of partials required when reassembling G from the node-fragments of a full decomposition tree of width k is at most*

$$\frac{[(k + 1)\Delta]!}{(\Delta!)^{k+1}} \cdot p((k + 1)\Delta) \cdot 2^{(k+1)\Delta}$$

That is, the number of partials has an upper bound that is independent of the number of vertices of the graph G .

Proof. As illustrated in Part 3 of the Running Example, we need to consider partials on $k + 1$ symbols. The isomorphism type of the k -vertex subgraph of amalgamation may change from amalgamation to amalgamation, and we provide for this by permitting the locations of the gaps to vary in the root-phrases. Since at most one gap occurs between two letters of the alphabet, the number of ways to insert the gaps is at most $2^{(k+1)\Delta}$. \square

NOTATION. For a subgraph-rooted graph (G, H) , we denote the corresponding set of partials by \mathcal{P}_H .

5 Embeddings of the Amalgamated Graph

DEF. Let ρ and σ be rotation systems for graphs X and Y , respectively, such that Y is a subgraph of X . We say that they are **consistent at a vertex** v of Y if the rotation σ at v is the restriction of the rotation ρ at v . If at every vertex of V_Y , the rotation σ is the restriction of the rotation ρ , then ρ and σ are **consistent rotation systems**. Moreover, the embeddings corresponding to rotations systems ρ and σ are then called **consistent embeddings**.

Given rotation systems ρ and σ for the rooted graphs (G, H) and (G', H') and an isomorphism $\eta : H \rightarrow H'$, such that the restriction of σ to $\eta(H)$ agrees with the restriction of σ to H' , we seek to count or give an upper bound for the number of embeddings of the amalgamated graph $(G *_\eta G', H *_\eta H')$ that are consistent with ρ and σ . The balance of this section is directed toward that objective. We consider the configuration at each vertex of the subgraph H .

A list of citations of early studies of restricted rotation systems is given by [24].

ISOLATED VERTICES IN THE SUBGRAPH H

The following proposition is to be used when there is a vertex of the root-subgraph H that has no neighbors in H .

Proposition 5.1. *Let (G, H) and (G', H') be subgraph-rooted graphs and $\eta : H \rightarrow H'$ an isomorphism. Let ρ and σ be rotation systems for G and G' such that $\sigma = \eta \circ \rho \circ \eta^{-1}$. Let v be an isolated vertex of the root-subgraph H , that is, with $\delta_H(v) = 0$. Then the number of rotations at v of $(G *_\eta G', H *_\eta H')$ that are consistent with ρ and σ is*

$$\delta_G(v) \binom{\delta_G(v) + \delta_{G'}(v) - 1}{\delta_G(v)} \tag{5.1}$$

Proof. A rotation at vertex v in $(G *_\eta G', H *_\eta H')$ is a cyclic ordering of the edges of $(G *_\eta G', H *_\eta H')$ that are incident at v . We regard the edges of E_G incident on vertex v as partitioning a cycle into $\delta_G(v)$ compartments into which edges of $E_{G'}$ incident at v are to be inserted. Once one of these $\delta_G(v)$ compartments is selected as the location of some arbitrarily selected “first” edge of $E_{G'}$, there are $\delta_G(v) + 1$ compartments into which the remaining $\delta_{G'}(v) - 1$ edges can be inserted. Since the order of these remaining edges is fixed, we may regard each of them as a zero, and partition them with $\delta_G(v)$ ones. Therefore, the number of ways to insert these $\delta_{G'}(v) - 1$ edges equals the number of binary strings of length $\delta_G(v) + \delta_{G'}(v) - 1$ with $\delta_G(v)$ ones. \square

Example 5.1. In [10], we amalgamate two vertex-rooted graphs (G, v) and (G', v') at 2-valent roots. Thus, $\delta_G(v) = \delta_{G'}(v) = 2$, and $\delta_H(v) = 0$. The number of rotations in the amalgamated graph that are consistent with a pair of rotations, one from G and one from G' , is

$$\delta_G(v) \binom{\delta_G(v) + \delta_{G'}(v) - 1}{\delta_G(v)} = 2 \binom{2 + 2 - 1}{2} = 2 \binom{3}{2} = 6$$

VERTICES OF POSITIVE DEGREE IN THE SUBGRAPH H

The case in which a vertex v of the root-subgraph H has positive degree requires sufficiently complicated notation, that it is useful to precede the general analysis by a definition, a lemma, and an example.

DEF. Let α and β be linear orderings of disjoint sets $S = \{x_1, \dots, x_p\}$ and $T = \{y_1, \dots, y_q\}$, respectively. We say that a linear ordering γ of the set $S \cup T$ is an **interleaving of the orderings** α and β if the restrictions of γ to S and T are identical to α and β , respectively.

Lemma 5.2. The number of ways to interleave two sequences of respective lengths p and q is

$$\binom{p + q}{q}$$

Proof. There is an obvious bijection from the set of interleavings to the set of binary strings of length $p + q$ with q ones. \square

Example 5.2. We depict in Figure 5.1 an amalgamation $G * G'$ in which the vertex v of the root-subgraph H has degree 2 in H , degree 7 in G , and degree 5 in G' . The rotations at vertex v are

$$\begin{aligned} \rho \cap \sigma \text{ in } H &: e_1, e_2 \\ \rho \text{ in } G &: c_1^1, c_2^1, c_3^1, e_1, c_1^2, c_2^2, e_2 \\ \sigma \text{ in } G' &: d_1^1, d_2^1, e_1, d_1^2, e_2 \end{aligned}$$

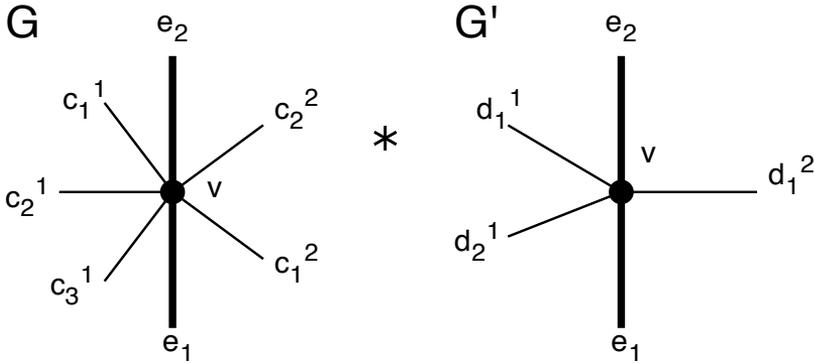


Figure 5.1: A vertex v of degree 2 in the (darkened) subgraph H , degree 7 in G , and degree 5 in G' .

The vertex v has degree 10 in the amalgamated graph $G * G'$. A rotation at v in $G * G'$ is consistent with the rotations ρ and σ if and only if the edge-sets $\{c_1^1, c_2^1, c_3^1\}$ and $\{d_1^1, d_2^1\}$ are interleaved between e_2 and e_1 and the edge-sets $\{c_1^2, c_2^2\}$ and $\{d_1^2\}$ are interleaved between e_1 and e_2 . By Lemma 5.2, the number of rotations at v in $G * G'$ that are consistent with the rotations ρ and σ is

$$\binom{5}{2} \binom{3}{1}$$

Proposition 5.3. Let (G, H) and (G', H') be subgraph-rooted graphs and $\eta : H \rightarrow H'$ an isomorphism. Let ρ and σ be rotation systems for G and G' such that $\sigma = \eta \circ \rho \circ \eta^{-1}$. Let v be a vertex of H with degree $\delta_H(v) > 0$ and the following rotations:

$$\begin{aligned} (\rho \cap \sigma)|_v \text{ in } H &: e_1, e_2, \dots, e_{\delta_H(v)} \\ \rho|_v \text{ in } G &: c_1^1, c_2^1, \dots, c_{\delta_1}^1, e_1, c_1^2, c_2^2, \dots, c_{\delta_2}^2, e_2, \dots, e_{\delta_H(v)} \\ \sigma|_v \text{ in } G' &: d_1^1, d_2^1, \dots, d_{\delta'_1}^1, e_1, d_1^2, d_2^2, \dots, d_{\delta'_2}^2, e_2, \dots, e_{\delta_H(v)} \end{aligned}$$

Then the number of rotations at v in $(G *_{\eta} G', H *_{\eta} H')$ that are consistent with ρ and σ is

$$\prod_{i=1}^{\delta_H(v)} \binom{\delta_i + \delta'_i}{\delta_i} \tag{5.2}$$

Proof. A rotation $\tau|_v$ at v in $(G *_\eta G', H *_\eta H')$ is consistent with the rotations $\rho|_v$ and $\sigma|_v$ if and only if it has the following property:

In the rotation $\tau|_v$, between two consecutive edges e_{i-1} and $e_i \in E_H$ (and between $e_{\delta_H(v)}$ and e_1) in the rotation $\rho \cap \sigma$, the edge sequences $c_1^i, c_2^i, \dots, c_{\delta_i}^i$ and $d_1^i, d_2^i, \dots, d_{\delta'_i}^i$ are interleaved.

By Lemma 5.2, the conclusion follows. □

COUNTING CONSISTENT ROTATION SYSTEMS

We have now arrived at the goal of this section.

Theorem 5.4. *Let (G, H) and (G', H') be subgraph-rooted graphs and $\eta : H \rightarrow H'$ an isomorphism. Let ρ and σ be rotation systems for G and G' such that $\sigma = \eta \circ \rho \circ \eta^{-1}$. For each vertex $v \in V_H$, we have the rotation*

$$(\rho \cap \sigma)|_v \text{ in } H : e_1^v, e_2^v, \dots, e_{\delta_H(v)}^v$$

according to the following parameter values:

- $\delta_H(v)$ is the degree of v in H .
- If $\delta_H(v) \neq 0$, then for $j = 1, \dots, \delta_H(v)$, let δ_j and δ'_j be the numbers of edges of E_G and $E_{G'}$, respectively, that lie between the edges e_{j-1} and e_j in the rotation $\rho|_v$ and in the rotation $\sigma|_v$, respectively (where $j-1$ is taken to be n if $j = 1$).

Then the number of rotation systems for $(G *_\eta G', H *_\eta H')$ that are consistent with ρ and σ is

$$\prod_{\substack{v \in V_H \\ \delta_H(v)=0}} \delta_G(v) \binom{\delta_G(v) + \delta_{G'}(v) - 1}{\delta_G(v)} \cdot \prod_{\substack{v \in V_H \\ \delta_H(v)>0}} \prod_{i=1}^{\delta_H(v)} \binom{\delta_i(v) + \delta'_i(v)}{\delta_i(v)} \quad (5.3)$$

Proof. This follows from Proposition 5.1 and Proposition 5.3. □

Theorem 5.5. *Let \mathcal{F} be a family of graphs of treewidth k and maximum degree Δ , each of which is rooted on an isomorphic copy of a spanning subgraph of K_{k+1} . Let $(G, H), (G', H') \in \mathcal{F}$, and let $\eta : H \rightarrow H'$ be an isomorphism. Let ρ and σ be rotation systems for G and G' such that $\sigma = \eta \circ \rho \circ \eta^{-1}$. Then the number of rotation systems for $(G *_\eta G', H *_\eta H')$ that are consistent with both ρ and σ is at most*

$$\left[\Delta \binom{2\Delta - 1}{\Delta} \right]^{k+1} \quad (5.4)$$

That is, it has an upper bound that is independent of the number of vertices of the graph G .

Proof. This theorem is a corollary of Theorem 5.4. □

SPECIAL CASE: AMALGAMATING ACROSS AN EDGE

The simplest case of amalgamating graphs (G, H) and (G', H') across isomorphic root-subgraphs with edges is the case in which the root-subgraphs are isomorphic to K_2 . The subcase in which both endpoints of both root-edges are 2-valent was developed in [18].

Corollary 5.6. *Let ρ and σ be rotation systems for the rooted graphs (G, vw) and $(G', v'w')$, where the pasting matches vertices v and w with vertices v' and w' , respectively. Then the number of embeddings of $(G *_{\eta} G', vw)$ that are consistent with ρ and σ is*

$$\binom{\deg_G v + \deg_{G'} v' - 2}{\deg_G v - 1} \binom{\deg_G w + \deg_{G'} w' - 2}{\deg_G w - 1} \tag{5.5}$$

Proof. Every embedding of $(G *_{\eta} G', vw)$ that is consistent with ρ and σ has its rotations completely determined by ρ and σ , except at the image of the vertices v and v' and at the image of the vertices w and w' . Figure 5.2 illustrates the situation. The names v and w for vertices in $G *_{\eta} G'$ adhere to the notational convention introduced in §3.

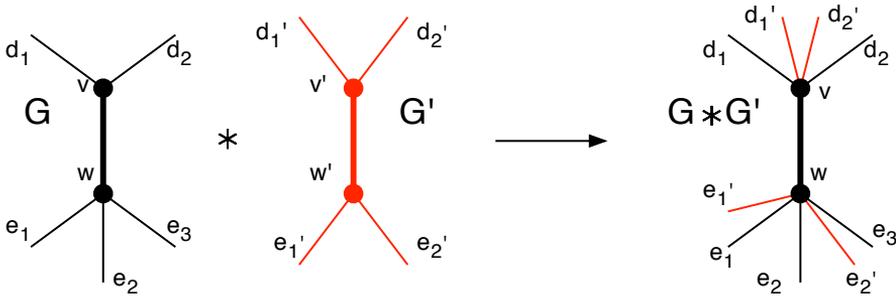


Figure 5.2: Amalgating two graphs on an edge.

Formula (5.5) follows from application of Theorem 5.4. □

REMARK. In Corollary 5.6, the notation vw for the root-edge of the graph $G *_{\eta} G'$ is consistent with the notational convention given in §3 for naming the root-vertices and root-edges of $G *_{\eta} G'$.

6 Production Rules for Graph Amalgamations

DEF. A **production** for a graph operation is a rule that prescribes the effect on the genus distribution of applying that operation to its operands. That is, based on the partials of the operands and the genera of their respective embedding surfaces, it says how many embeddings the resulting graph has of each genus and of each type of partial. The operands and the operation appear at the tail of a right-arrow and are called the **antecedent of the production**. The consequential information appears at the head of the right-arrow and is called the **consequent of the production**.

Running Example – Part 4. A single production suffices to calculate the partial genus distribution (4.3) of the compound fragment $G[h; b, g; e]$ from partial genus distributions (4.1) and (4.2), for the node-fragments $G[; e, g; h]$ and $G[; e, g; b]$, respectively:

$$(egh, ehg)_i * (egb, ebg)_j \longrightarrow [eg; eh, gh; eb, gb] \\ (be\bullet g, bge, eg\bullet)_{i+j} + (bg\bullet e, beg, e\bullet g)_{i+j} \\ + (beg\bullet ebg\bullet g)_{i+j+1} + (be\bullet g\bullet ebg\bullet eg)_{i+j+1}$$

REMARK. We observe that this production is asymmetric. That is, it converts the pivot vertex h of the earlier node-fragment in the post-order into a bullet.

This particular form of production is designed for reassembling a graph from a full decomposition tree by iterative amalgamation of fragments. In each case, all of the vertices of each of the two amalgamand-fragments lie in a single node within that fragment. For treewidth k , the two nodes intersect in k vertices. We observe that the subscript

$$[eg; eh, gh; eb, gb]$$

of the arrow operator contains three lists of edges.

1. The first list is the set of edges that lie in the root-subgraphs of both amalgamand fragments.
2. The second list contains the other edges that join vertices within the presenting node of the first amalgamand fragment.
3. The third list contains the other edges that join vertices within the presenting node of the second amalgamand fragment.

The antecedent of each production is a root-phrase with an integer variable as subscript, followed by an asterisk denoting amalgamation, and then another root-phrase with another integer variable as subscript. The consequent of each production is a sum of root-phrases, each of which has as its subscript a formula giving the genus of the surface of the amalgamated graph that corresponds to the two operand embeddings.

The coefficient of each such subscripted partial indicates in how many ways an fb-walk corresponding to that partial can be created by amalgamating two operand embeddings that correspond to the two partials in the antecedent. The sum of the coefficients is subject to the upper bound of Formula 5.3.

If the two root-phrases in the antecedent are inconsistent on the root-subgraph (which can occur for treewidth four or more), then the consequent is empty. We use the two partials in the antecedent and the first list of edges in the subscript of the arrow operator to construct rotation systems for the set of embeddings of the amalgamated graph that are consistent with those two partials. Figure 6.1 illustrates the situation for Running Example – Part 4. For each of these rotation systems, we use the Heffter-Edmonds face-tracing algorithm to calculate the number of faces in the consequent embeddings relative to the sum of the numbers in the antecedent embeddings, after which we use the Euler polyhedral equate to calculate the corresponding increment or decrement in genus. The time required depends only on the root-subgraphs, not on the numbers of vertices in the amalgamands.

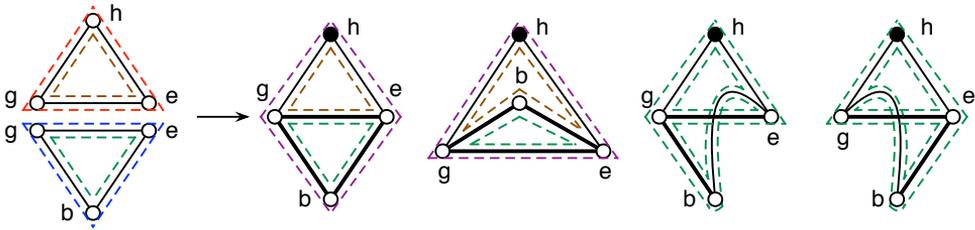


Figure 6.1: Deriving a production.

Theorem 6.1. *Let \mathcal{G} be a family of subgraph-rooted graphs of treewidth at most k and maximum degree at most Δ , where k and Δ are fixed positive integers. Let $(G, H), (G', H') \in \mathcal{G}$, and let $\eta : H \rightarrow H'$ be an isomorphism. Then the number of productions required to calculate the pgd of $(G *_\eta G', H)$ from the pgd's of (G, H) and (G', H') is at most*

$$\left(\frac{[(k+1)\Delta]!}{(\Delta!)^{k+1}} \cdot p((k+1)\Delta) \cdot 2^{(k+1)\Delta} \right)^2$$

That is, the number of productions has an upper bound that is independent of the numbers of vertices of the graphs G and G' .

Proof. One needs a production for each ordered pair of partials used, and no more. Thus, the number of productions needed is at most the square of the number of partials determined by Theorem 4.2. □

RECURRENCES AND CLOSED FORMS

Productions were used in [10], and then in [7], [18], and [15] to derive systems of simultaneous recurrence relations for pgds. Using generating functions on the simultaneous recurrences, one can sometimes derive closed forms, as in [5].

7 Genus Distribution Algorithm

A skeletal version of the algorithm for calculating the genus distribution of the graphs in a family \mathcal{G} of graphs of treewidth k and maximum degree at most $\Delta_{\mathcal{G}}$ is straightforward and intuitive.

Algorithm 7.1 (Genus Distribution Algorithm).

Input: an n -vertex graph G of treewidth k and maximum degree Δ

Output: a pgd and the genus distribution for G .

Comment: INITIALIZE

1. Calculate a full decomposition tree T of width k for G .

2. Determine a post-order for decomposition-tree T , with the lines $\ell_1, \ell_2, \dots, \ell_{n-k-1}$ in the post-order, so that the j^{th} amalgamation is across line ℓ_j .
3. For $j = 1, \dots, n - k - 1$,
 - (a) Construct the subgraph-rooted node-fragment (M_j, M_j, u_j) as the induced graph on the j^{th} node of T in the post-order.
 - (b) Identify the partner (M'_j, M'_j, u'_j) , and construct the isomorphism $\eta_j : M_j - \{u_j\} \rightarrow M'_j - \{u'_j\}$
 Comment: “Partner” is defined in §3.
 - (c) Calculate the pgd of the node-fragment (M_j, M_j, u_j) .
4. Let $(F_1, H_1, u_1) = (M_1, M_1, u_1)$, and let (F'_1, H'_1, u'_1) be its partner, i.e., let $(F'_1, H'_1, u'_1) = (M'_1, M'_1, u'_1)$

Comment: MAIN LOOP REASSEMBLES G

5. For each line ℓ_j of the decomposition tree
 - (a) Amalgamate the two fragments (F_j, H_j, u_j) and (F'_j, H'_j, u'_j) via the isomorphism η_j to produce the fragment F_{j+1} .
 - (b) Let H_{j+1} be whichever of M_j or M'_j is the latter node-fragment in the post-order. Let the pivot u_{j+1} be whichever vertex of the root-subgraph H_{j+1} will not be pasted in the next amalgamation that involves the fragment F_{j+1} .
 - (c) Use productions to calculate the pgd of $(F_{j+1}, H_{j+1}, u_{j+1})$ from the pgd's of (F_j, H_j, u_j) and (F'_j, H'_j, u'_j) .

Comment: CALCULATE g_i 's BY SUMMING.

N.B. We recall from §4 that for a subgraph-rooted graph (G, H) , we denote the corresponding set of partials by \mathcal{P}_H .

6. For $i = 0, \dots, \lfloor \beta(G)/2 \rfloor$, calculate $g_i(G)$ by the formula

$$g_i(G) = \sum_{\pi \in \mathcal{P}_{H_{n-k}}} \pi_i(G)$$

BREADTH OF THE GENUS RANGE

Analysis of Algorithm 7.1 uses the concept of *breadth of the genus range*.

DEF. The *breadth of the genus range* of a graph G is given by the equation

$$\gamma_B(G) = \gamma_{\max}(G) - \gamma_{\min}(G) + 1$$

Thus, $\gamma_B(G)$ is equal to the number of values of the index i such that $g_i(G) \geq 1$.

Proposition 7.1. *The breadth of the genus range of any graph G is at most $(2|E_G| + 3)/6$.*

Proof.

$$\begin{aligned}
 \gamma_B(G) &= \gamma_{\max}(G) - \gamma_{\min}(G) + 1 \\
 &\leq \frac{\beta(G)}{2} - \gamma_{\min}(G) + 1 \\
 &\leq \frac{|E_G| - |V_G| + 1}{2} - \frac{|E_G| - 3|V_G| + 6}{6} + 1 \\
 &= \frac{2|E_G| + 3}{6}
 \end{aligned}$$

□

Corollary 7.2. *Let \mathcal{G} be a family of simple graphs of maximum degree at most $\Delta_{\mathcal{G}}$. Then the breadth of the genus range of any n -vertex graph G in \mathcal{G} is at most $(n\Delta_{\mathcal{G}} + 3)/6$.*

Proof. Since $2|E_G| \leq n\Delta_{\mathcal{G}}$, by Euler’s theorem on degree sum, this follows immediately from Proposition 7.1. □

ANALYSIS OF THE ALGORITHM

As established by [26], assigning an upper bound to the degree of a graph does not reduce the computational complexity of the minimum genus problem. Thus, bounding the treewidth is essential to the computational complexity of our algorithm.

Proposition 7.3. *Let \mathcal{G} be a family of subgraph-rooted graphs, each of treewidth at most k and of maximum degree at most Δ , where k and Δ are fixed positive integers. Let $(G, H, u), (G', H', u') \in \mathcal{G}$ and let $\eta : H - \{u\} \rightarrow H' - \{u'\}$ be an isomorphism. Then the time required to calculate the pgd of $(G *_{\eta} G', H, u)$ from the pgd’s of (G, H, u) and (G', H', u') is in $O(|V_G| \cdot |V_{G'}|)$.*

Proof. The time depends predominantly on the number of applications of productions, which depends, in turn, only on the product of the numbers of partials in the respective pgd’s of G and G' with non-zero subscripted coefficients. The number of non-zero subscripted coefficients of the partials depends on the breadths of their genus ranges. This theorem now follows from Corollary 7.2 and Theorem 4.2. □

Theorem 7.4. *Let \mathcal{G} be a family of graphs of treewidth at most k and maximum degree at most Δ , where k and Δ are fixed positive integers. Then the time needed to calculate the genus distribution of an n -vertex graph $G \in \mathcal{F}$, starting from a full decomposition tree T for G , is in $O(n^2)$.*

Proof. We proceed step by step.

Step 1. A full decomposition tree T for the graph G can be calculated in time proportional to $|V_T|$, which is in $O(n)$, by Corollary 2.7.

Step 2. The post-order of the decomposition tree T can be calculated in time proportional to $|V_T|$, which is in $O(n)$.

Substep 3a. The time needed to construct each of the node-fragments M_1, \dots, M_{n-k} from the given graph G is in $O(k^2)$, since each node of the decomposition tree T has $k + 1$ vertices. The time for this substep is in $O(1)$.

Substep 3b. Each partner can be located by a post-order traversal. Each isomorphism η_j is an artifact of the tree decomposition. The time for this substep is in $O(1)$.

Substep 3c. There is a fixed upper bound of $(k!)^{k+1}$ for the number of embeddings of M_j , since M_j is a subgraph of the complete graph K_{k+1} . The time needed to determine the appropriate subscripted partial for each embedding of a node-fragment, by using the Heffter-Edmonds algorithm, is in $O(k\Delta_{M_j})$, that is, linear in the number of edges of the node-fragment, and constant, accordingly, with respect to $|V_G|$.

Step 3 total. Since there are $n - k$ node-fragments, by Proposition 2.5, it follows that the total time needed for Step 3 is in $O(n)$.

Step 4. This initializing step takes $O(1)$ -time.

Substep 5a. The time needed to calculate a representation of the amalgamated graph F_{j+1} from representations of the amalgamands is proportional to $|E_{F_j}| + |E_{F'_j}|$, which is linear in $|V_{F_j}| + |V_{F'_j}|$, because the degree of the graph G is bounded. This substep takes $O(|V_{F_j}| + |V_{F'_j}|)$ -time.

Substep 5b. This substep is achieved by referring to the post-order traversal of tree T .

Substep 5c. The number of productions is constant, by Theorem 6.1. The numbers of subscripted partials with non-zero coefficient in the pgd's of the amalgamand fragments are proportional to $\gamma_B(F_j)$ and $\gamma_B(F'_j)$, respectively. By Corollary 7.2, $\gamma_B(F_j)$ and $\gamma_B(F'_j)$ are proportional to $|V_{F_j}|$ and $|V_{F'_j}|$, respectively. By Proposition 7.3, the time for this substep is $O(|V_{F_j}| \cdot |V_{F'_j}|)$, subject to the assumption that multiplication of integers takes constant time.

Step 5 total. Let \mathcal{S} denote the sequence of pairs of fragments that occurs in the reassembly of G from the node-fragments corresponding to the decomposition tree T . The time for each iteration of the body of the loop in Step 4 is dominated by the time needed for Substep 5c. Thus, the total time needed to calculate the pgd of G is at most

$$\sum_{(F,F') \in \mathcal{S}} c |V_F| \cdot |V_{F'}| = c \sum_{(F,F') \in \mathcal{S}} |V_F| \cdot |V_{F'}|$$

We further suppose that the number of vertices in the node-fragment F_i is n_i , for $i = 1, \dots, n - k$, and we observe that in the total reassembly no two node-fragments occur

twice as subgraphs of fragments that are amalgamated. It follows that

$$\begin{aligned} \sum_{(F,F') \in \mathcal{S}} |V_F| \cdot |V_{F'}| &= \sum_{i=1}^{n-k} \sum_{j=1}^{n-k} n_i n_j - \sum_{i=1}^{n-k} n_i^2 \\ &< (n_1 + n_2 + \dots + n_{n-k})^2 \\ &= n^2 \end{aligned}$$

Step 6. Of course, each term $g_i(G)$ in the genus distribution for G is obtained from the pgd by summing all the coefficients of subscripted partials with subscript i . The time needed for such a calculation of each $g_i(G)$ is proportional to the number of different partials, which is bounded by a constant. Thus, the time needed to calculate the genus distribution from the pgd is proportional to the size of the genus range, and thus, in $O(n)$. \square

8 Conclusions

Stahl [22, 23] has called a family of graphs *H-linear* if its members can be derived by iterative amalgamation of copies of a graph H , and he has introduced a form of production matrices whose elements are univariate polynomials, in which the degree of a term corresponds to an increment of genus as an additional copy of the graph H is amalgamated to a growing linear chain. The treewidth of the graphs in an H -linear family is the tree width of the graph H . Although the size of such matrices can be very large, corresponding to the number of partial genus distributions associated with a given maximum degree, it is of fixed size. Accordingly, the time-complexity needed to take a power of such a production matrix depends only on the time needed to multiply two polynomials of linearly increasing degree.

Practical algorithms for the genus distributions and partitioned genus distributions of the graphs in various interesting linear families of graphs, implicit in [23, 10, 15, 18, 11, 12], fall within the quadratic time-complexity upper limit given by Theorem 7.4. Moreover, quadratic-time calculation of genus distributions is implicit in [7, 19] for graph families, including circular ladders and Möbius ladders, that are not H -linear, but which could be characterized as *ring-like*. Beyond that, there is a practical quadratic-time algorithm for the cubic Halin graphs [9], which are a non-linear family.

We observe that in a genus distribution calculation by our algorithm, the partials and productions can be generated dynamically as needed, rather than in advance. This suggests the feasibility of implementing such an algorithm for graphs of reasonably small treewidth and maximum degree, since the number of productions needed might be far smaller than the total number possible for that treewidth and degree.

The exposition here illustrates once again how bounding the treewidth can be used to reduce otherwise NP-hard calculations regarding embeddings to polynomial time. Here we have also bounded the degree. This immediately suggests this general problem:

Research Problem 1. Determine whether the genus distribution of the graphs of bounded treewidth can be calculated in polynomial-time, if the degree is not bounded.

Of course, rather than being content with an algorithm with such a vast proliferation of partials and productions, one hopes for closed formulas or tractable recursions for interesting classes of graphs. A related line of investigation would involve bounding the treewidth and prescribing the minimum genus (which effectively bounds the average degree). The following two problems may be approachable.

Research Problem 2. Algorithms for calculating the genus distributions of 3-regular and 4-regular outerplanar graphs are given by [8] and [19], respectively. Calculate the genus distributions of arbitrary outerplanar graphs.

Research Problem 3. A *Halin graph* is obtained from a plane tree with at least four vertices and no vertices of degree two, by drawing a cycle through the leaves. An algorithm for the genus distribution of any 3-regular Halin graph is given by [9]. Calculate the genus distributions of arbitrary Halin graphs.

Research Problem 4. A remaining problem of self-evident theoretical interest is determination of a lower bound on the time needed to calculate the genus distribution of a graph of fixed treewidth and bounded maximum degree.

ACKNOWLEDGEMENT

The author wishes to thank Imran Khan and Mehvish Poshni for their many valuable comments on the early drafts of this paper. The author also wishes to thank an anonymous referee for various helpful suggestions.

References

- [1] L. W. Beineke, R. J. Wilson, J. L. Gross, and T. W. Tucker, Editors, *Topics in Topological Graph Theory*, Cambridge Univ. Press, 2009.
- [2] H. L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, *SIAM J. Comput.* **25** (1996), 1305–1317.
- [3] H. L. Bodlaender, A partial k -arboretum of graphs with bounded treewidth, *Theoretical Comp. Sci.* **209** (1998), 1–45.
- [4] R. B. Borie, R. G. Parker, and C. A. Tovey, Recursively constructed graphs, §2.4 (pp. 101–122) of *Handbook of Graph Theory*, second edition, eds. J. L. Gross, J. Yellen, and P. Zhang, CRC Press (imprint of Taylor and Francis), 2014.
- [5] M. L. Furst, J. L. Gross and R. Statman, Genus distribution for two classes of graphs, *J. Combin. Theory (B)* **46** (1989), 22–36.

- [6] J. L. Gross, Genus distribution of graphs under surgery: adding edges and splitting vertices, *New York J. Math.* **16** (2010), 161–178.
- [7] J. L. Gross, Genus distribution of graph amalgamations: Self-pasting at root-vertices, *Australasian J. Combin.* **49** (2011), 19–38.
- [8] J. L. Gross, Genus distributions of cubic outerplanar graphs, *J. of Graph Algorithms and Applications* **15** (2011), 295–316.
- [9] J. L. Gross, Embeddings of cubic Halin graphs: a surface-by-surface inventory, *Ars Math. Contemporanea* **6** (2013), 37–56.
- [10] J. L. Gross, I. F. Khan, and M. I. Poshni, Genus distribution of graph amalgamations: Pasting at root-vertices, *Ars Combin.* **94** (2010), 33–53.
- [11] J. L. Gross, I. F. Khan, and M. I. Poshni, Genus distribution for iterated claws, preprint 2013, 20pp.
- [12] J.L. Gross, T. Mansour, T.W. Tucker, and D.G.L. Wang, Log-concavity of combinations of sequences and applications to genus distributions, draft manuscript, 2013.
- [13] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Dover, 2001; (original edn. Wiley, 1987).
- [14] K. Kawarabayashi, B. Mohar, and B. Reed, A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width, *Proc. 49th Ann. Symp. on Foundations of Computer Science (FOCS'08)* IEEE (2008), 771–780.
- [15] I. F. Khan, M. I. Poshni, and J. L. Gross, Genus distribution of graph amalgamations at roots of higher degree, *Ars Math. Contemporanea* **3** (2010), 121–138.
- [16] I. F. Khan, M. I. Poshni, and J. L. Gross, Genus distribution of $P_3 \square P_n$, *Discrete Math.* **312** (2012), 2863–2871.
- [17] B. Mohar and C. Thomassen, *Graphs on Surfaces*, Johns Hopkins University Press, 2001.
- [18] M. I. Poshni, I. F. Khan, and J. L. Gross, Genus distribution of edge-amalgamations, *Ars Math. Contemporanea* **3** (2010), 69–86.
- [19] M. I. Poshni, I. F. Khan, and J. L. Gross, Genus distribution of 4-regular outerplanar graphs, *Electronic J. Combin.* **18** (2011) #P212, 25pp.
- [20] M. I. Poshni, I. F. Khan, and J. L. Gross, Genus distribution of graphs under self-edge-amalgamations, *Ars Math. Contemporanea* **5** (2012), 127–148.
- [21] N. Robertson and P. D. Seymour, Graph minors. II, Algorithmic aspects of tree-width, *J. Algorithms* **7** (1986), 309–322.

- [22] S. Stahl, Permutation-partition pairs. III. Embedding distributions of linear families of graphs, *J. Combin. Theory, Ser. B* **52** (1991), 191–218.
- [23] S. Stahl, On the zeros of some genus polynomials, *Canad. J. Math.* **49** (1997), 617–640.
- [24] J. Širáň and M. Škoviera, Relative embeddings of graphs on closed surfaces, *Math. Nachr.* **136** (1988), 275–284.
- [25] C. Thomassen, The graph genus problem is NP-complete, *J. Algorithms* **10** (1989), 568–576.
- [26] C. Thomassen, The genus problem for cubic graphs, *J. Combin. Theory (B)* **69** (1997), 52–58.
- [27] J. A. Wald and C. J. Colbourn, Steiner trees, partial 2-trees, and minimum IFI networks, *Networks* **13** (1983), 159–167.
- [28] A. T. White, *Graphs of Groups on Surfaces*, North-Holland, 2001.