



HAL
open science

Dynamic Programming for Graphs on Surfaces

Juanjo Rué, Ignasi Sau, Dimitrios M. Thilikos

► **To cite this version:**

Juanjo Rué, Ignasi Sau, Dimitrios M. Thilikos. Dynamic Programming for Graphs on Surfaces. [Research Report] RR-7166, INRIA. 2009, pp.39. inria-00443582

HAL Id: inria-00443582

<https://inria.hal.science/inria-00443582>

Submitted on 30 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Programmation Dynamique pour les Graphes dans
les Surfaces*

Juanjo Rué — Ignasi Sau — Dimitrios M. Thilikos

N° 7166

December 2009

Thème COM

*R*apport
de recherche



Programmation Dynamique pour les Graphes dans les Surfaces

Juanjo Rué^{*}, Ignasi Sau^{†‡}, Dimitrios M. Thilikos[§]

Thème COM — Systèmes communicants
Projet MASCOTTE

Rapport de recherche n° 7166 — December 2009 — 39 pages

Résumé : Dans cet article nous construisons d'un *framework* pour résoudre une classe très large de problèmes d'optimisation dans les graphes plongeables dans une surface, en utilisant des algorithmes de programmation dynamique sur des décompositions en branches. Un graphe est *plongeable* dans une surface Σ s'il peut être dessiné dans Σ sans croisements d'arêtes. Les graphes planaires correspondent au cas quand Σ est la sphère. Informellement, une *décomposition en branches* d'un graphe est une décomposition de ce graphe sous forme d'un arbre. Ainsi la *largeur de branches* (en anglais, *branchwidth*) mesure sa "proximité" avec un arbre. Étant donné une décomposition en branches de largeur k d'un graphe à n sommets, de nombreux problèmes peuvent être résolus en temps $2^{O(k \cdot \log k)} \cdot n$ en utilisant programmation dynamique. Se "débarrasser" du facteur $\log k$ est très désirable mais ne pas toujours possible; typiquement G doit avoir "peu d'arêtes" et le problème ne doit pas être "trop compliqué". Les algorithmes en temps $2^{O(k)} \cdot n$ s'appellent *simple-exponentiels*.

Nous développons une approche pour concevoir des algorithmes simple-exponentiels pour la classe des graphes plongeables dans une surface et pour des problèmes tels que les *tables* de la programmation dynamique encodent partitions d'ensembles de sommets. Ce résultat élargi remarquablement la classe de problèmes simple-exponentiels et implique et améliore plusieurs algorithmes existants. L'ingrédient principal de l'approche est la définition des *surface cut decompositions*, une décomposition en branches pour les graphes dans les surfaces qui généralise les *sphere cut decompositions* définies par Seymour et Thomas pour les graphes planaires. La construction des *surface cut decompositions* utilise un outil topologique appelé *décomposition polyédrique*. Le résultat principal est que si la programmation dynamique se fait sur une *surface cut decomposition*, alors le temps d'exécution est simple-exponentiel. Pour prouver cette propriété, on utilise des techniques de la théorie topologique des graphes et de combinatoire analytique. En particulier, on étend les *structures de Catalan* (qui comptent les *non-crossing partitions* dans les graphes planaires) pour les graphes dans les surfaces.

^{*} Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Barcelona, Spain.

[†] Mascotte joint Project, INRIA/CNRS-I3S/UNS, 2004 route des Lucioles, B.P. 93 - F-06902, Sophia-Antipolis, France.

[‡] Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya, Barcelona, Spain.

[§] Department of Mathematics, National and Kapodistrian University of Athens, Greece.

Mots-clés : ALgorithmes paramétrés, combinatoire analytique, graphes dans les surfaces, largeur de branches, programmation dynamique, méthode symbolique, partitions non-croisés.

Dynamic Programming for Graphs on Surfaces

Abstract: We provide a framework for the design of $2^{O(k)} \cdot n$ step dynamic programming algorithms for surface-embedded graphs on n vertices of branchwidth at most k . Our technique applies to graph problems for which dynamic programming uses tables encoding set partitions. For general graphs, the best known algorithms for such problems run in $2^{O(k \cdot \log k)} \cdot n$ steps. That way, we considerably extend the class of problems that can be solved by algorithms whose running times have a *single exponential dependence* on branchwidth, and improve the running time of several existing algorithms. Our approach is based on a new type of branch decomposition called *surface cut decomposition*, which generalizes sphere cut decompositions for planar graphs, and where dynamic programming should be applied for each particular problem. The construction of such a decomposition uses a new graph-topological tool called *polyhedral decomposition*. The main result is that if dynamic programming is applied on surface cut decompositions, then the time dependence on branchwidth is *single exponential*. This fact is proved by a detailed analysis of how non-crossing partitions are arranged on surfaces with boundary and uses diverse techniques from topological graph theory and analytic combinatorics.

Key-words: Parameterized algorithms, analytic combinatorics, graphs on surfaces, branchwidth, dynamic programming, polyhedral embeddings, symbolic method, non-crossing partitions.

1 Introduction

One of the most important parameters in the design and analysis of graph algorithms is the branchwidth of a graph. Branchwidth, together with its twin parameter of treewidth, can be seen as a measure of the topological resemblance of a graph to a tree. Its algorithmic importance has its origins in the celebrated theorem of Courcelle (see e.g. [8]), stating that graph problems expressible in Monadic Second Order Logic can be solved in $f(\mathbf{bw}) \cdot n$ (here \mathbf{bw} is the branchwidth¹ and n is the number of vertices of the input graph). Using the parameterized complexity terminology, this implies that a huge number of graph problems are fixed-parameter tractable when parameterized by the branchwidth of their input graph. As the bounds for $f(\mathbf{bw})$ provided by Courcelle’s theorem are huge, the design of tailor-made dynamic programming algorithms for specific problems, so that $f(\mathbf{bw})$ is a simple (preferably single exponential) function, became a natural (and unavoidable) ingredient for many papers on algorithms design (see [2, 5, 12, 29]).

Dynamic programming. Dynamic programming is applied in a bottom-up fashion on a rooted branch decomposition of the input graph, that roughly is a way to decompose the graph into a tree structure of edge bipartitions (the formal definition is in Section 2). Each bipartition defines a separator of the graph called *middle set*, of cardinality bounded by the branchwidth of the input graph. The decomposition is routed in the sense that one of the parts of each bipartition is the “lower part of the middle set”, i.e. the so-far processed one. For each graph problem, dynamic programming requires the suitable definition of tables encoding how potential (global) solutions of the problem are restricted in the middle set and the corresponding lower part. The size of these tables reflects the dependence of \mathbf{bw} in the running time of the dynamic programming. Defining the tables is not always an easy task, as they depend on the particularities of each problem (some typical examples are shown in Section 3). In many cases, problems are grouped together according to the similarities in the way to treat them, and usually this leads to distinct upper bounds for the function $f(\mathbf{bw})$. We define the following categories of dynamic programming algorithms (below S denotes a the middle set of a branch decomposition) :

- (A) those where the tables encode a fixed number of *vertex subsets of S* ;
- (B) those where the tables encode a fixed number of *connected pairings of vertices of S* ; and
- (C) those where the tables encode a fixed number of *connected packings of S into sets*.

In Categories (B) and (C), by the term *connected* for the pairings (resp. packings) we mean that they correspond to a packing of paths (resp. trees) in the lower part of the middle set S . The above classification also induces a classification of graph problems depending on whether they can be solved by some algorithm in some of the above categories. To facilitate our presentation, we present in Section 3 the dynamic programming algorithms for a problem in Category (A) and a problem in Category (C). Notice that the problems in Category (A) belong also to Category (B), and problems in Category (B) are also problems in Category (C). Clearly, the size of the tables for problems in Category (A) is a single exponential function of the middle set size. Therefore, for such problems

¹The original statement of Courcelle’s theorem used the parameter of treewidth instead of branchwidth. The two parameters are approximately equivalent, in the sense that the one is a constant factor approximation of the other [25].

we have that $f(\mathbf{bw}) = 2^{O(\mathbf{bw})}$. Such problems are, for instance, 3-COLORING, VERTEX COVER, DOMINATING SET, or INDEPENDENT SET, whose common characteristic is that the certificate of the solution is a set (or a fixed number of sets) of vertices whose choice is not restricted by some global condition. Unfortunately, when connectivity conditions are applied, the tables of the dynamic programming are of size $2^{O(\mathbf{bw} \cdot \log(\mathbf{bw}))}$ or more. This happens because one needs to encode more information on the way a possible solution of the problem is situated in the middle set S , which usually classifies the problems in categories (B) or (C). Typical problems in Category (B) are LONGEST PATH and HAMILTONIAN CYCLE, where pairings correspond to the connected portions of a solution to the lower part of the middle set. Typical problems in Category (C) are CONNECTED VERTEX COVER and MAXIMUM INDUCED FOREST², where the connected portions of a solution may be identified by sets of arbitrary cardinality. For Category (B), the size of the tables is lower-bounded by the number of perfect matchings of a complete bipartite graph of k vertices, that is by $2^{\Theta(k \cdot \log k)}$. For Category (C), the size of the tables is lower bounded by the k -th Bell number, that is again lower-bounded by $2^{\Theta(k \cdot \log k)}$. In both cases, this implies algorithms where $f(\mathbf{bw}) = 2^{O(\mathbf{bw} \cdot \log \mathbf{bw})}$.

Single-exponentiality : results and techniques. The most desired characteristic of any dynamic programming algorithm is the single exponential dependence on the branchwidth of the input graph (according to the results in [23], this dependence is optimal for many combinatorial problems). Exponential dependence is trivial for problems in Category (A), and may become possible for the other two categories when we take into account the *structural properties* of the input graph. For planar graphs, the first step in this direction was done in [14] for problems in Category (B) and certain problems in Category (C) such as PLANAR TSP (see [27] for extensions of this technique for more problems in Category (C)). The idea in [14] is to use a special type of branch decomposition called *sphere cut* decomposition (introduced in [28]) that can guarantee that the pairings are non-crossing pairings (because of the connectivity demand) around a virtual edge-avoiding cycle (called *noose*) of the plane where G is embedded. This restricts the number of tables corresponding to a middle set of size k by the k -th Catalan number, which is *single-exponential* in k . The same approach was extended for graphs of Euler genus γ in [11] for problems in Category (B). The idea was to perform a *planarization* of the input graph by splitting the potential solution into at most γ pieces and then applying the sphere cut decomposition technique of [14] to a more general version of the problem where the number of pairings is still bounded by some Catalan number. This made it possible to avoid dealing with the combinatorial structures in surfaces, where the arrangement of the solutions are harder to handle. The same idea was applied in [13] for H -minor free graphs using much more involved Catalan structures, again for problems in Category (B).

Our results. In this paper, we follow a different approach in order to design single exponential (in \mathbf{bw}) algorithms for graphs embedded in surfaces. In particular, we deviate significantly from the planarization technique of [11]. Instead, we extend the concept of sphere cut decomposition from planar graphs to surfaces and we exploit directly the combinatorial structure of the potential solutions in the topological surface. Our approach permits us to provide combinatorial bounds for problems

²Notice that the MAXIMUM INDUCED FOREST problem is equivalent to the FEEDBACK VERTEX SET problem. We choose this way to present it in order to make more visible its classification into Category (C).

in Category (C). Apart from those mentioned above, examples of such problems are MAXIMUM d -DEGREE-BOUNDED CONNECTED SUBGRAPH, MAXIMUM d -DEGREE-BOUNDED CONNECTED INDUCED SUBGRAPH and all the variants studied in [27], CONNECTED DOMINATING SET, CONNECTED r -DOMINATION, CONNECTED FVS, MAXIMUM LEAF SPANNING TREE, MAXIMUM FULL-DEGREE SPANNING TREE, MAXIMUM EULERIAN SUBGRAPH, STEINER TREE, and MAXIMUM LEAF TREE. As Category (C) includes the problems in Category (B), our results imply all the results in [11], and with running times whose genus dependence is better than the ones in [11], as discussed in Section 8.

Our techniques. Our analysis is based on a special type of branch decomposition of embedded graphs with nice topological properties, which we call *surface cut decomposition* (see Section 6). Roughly, the middle sets of such a decomposition are situated along a bounded (by the genus γ) set of nooses of the surface with few (again bounded by γ) common points. The construction of such a decomposition is based on the concept of *polyhedral decomposition* introduced in Section 4. In Section 5, we prove some basic properties of surface cut decompositions that make it possible to bound the sizes of the tables of the dynamic programming. They correspond to the number of non-crossing partitions of vertex sets laying in the boundary of a generic surface. To count these partitions, we use a powerful technique of analytic combinatorics : *singularity analysis* over expressions obtained by the *symbolic method* (for more on this technique, see the monograph of Flajolet and Sedgewick [16]). The symbolic method gives a precise asymptotic enumeration of the number of non-crossing partitions, that yields the single exponentiality of the table size (see Section 7). To solve a problem in Category (C), our approach resides on a common preprocessing step that is to construct the *surface cut decomposition* (Algorithm 2 in Section 6). Then, what remains is just to run the dynamic programming algorithm on such a surface cut decomposition. The exponential bound on the size of the tables of this dynamic programming algorithm is provided as a result of our analysis in Theorem 8.1 of Section 8. We first provide the necessary preliminaries and examples of dynamic programming algorithms in Sections 2 and 3, respectively.

2 Preliminaries

Sections 2.1, 2.2, 2.3, and 2.4 contain the basic background and the notation we will use concerning topological surfaces, graphs embedded in surfaces, tree-like decompositions of graphs, and the symbolic method and analytic combinatorics, respectively.

2.1 Topological surfaces

In this article, surfaces are compact and their boundary is homeomorphic to a finite set (possibly empty) of disjoint circles. We denote by $\beta(\Sigma)$ the number of connected components of the boundary of a surface Σ . The Surface Classification Theorem [24] asserts that a compact and connected surface without boundary is determined, up to homeomorphism, by its Euler characteristic $\chi(\Sigma)$ and by whether it is orientable or not. More precisely, orientable surfaces are obtained by adding $g \geq 0$ *handles* to the sphere \mathbb{S}^2 , obtaining the g -torus \mathbb{T}_g with Euler characteristic $\chi(\mathbb{T}_g) = 2 - 2g$, while

non-orientable surfaces are obtained by adding $h > 0$ *cross-caps* to the sphere, hence obtaining a non-orientable surface \mathbb{P}_h with Euler characteristic $\chi(\mathbb{P}_h) = 2 - h$. We denote by $\bar{\Sigma}$ the surface (without boundary) obtained from Σ by gluing a disk on each of the $\beta(\Sigma)$ components of the boundary. It is then easy to show that $\chi(\bar{\Sigma}) = \beta(\Sigma) + \chi(\Sigma)$. A subset Π of a surface Σ is *surface-separating* if $\Sigma \setminus \Pi$ has at least 2 connected components.

As a conclusion, our surfaces are determined, up to homeomorphism, by their orientability, their Euler characteristic and the number of connected components of their boundary. For computational simplicity, it is convenient to work with the *Euler genus* $\gamma(\Sigma)$ of a surface Σ , which is defined as $\gamma(\Sigma) = 2 - \chi(\Sigma)$.

2.2 Graphs embedded in surfaces

Our main reference for graphs on surfaces is the monograph of Mohar and Thomassen [24]. For a graph G we use the notation (G, τ) to denote that τ is an embedding of G in Σ , whenever the surface Σ is clear from the context. An embedding has *vertices*, *edges*, and *faces*, which are 0, 1, and 2 dimensional open sets, and are denoted $V(G)$, $E(G)$, and $F(G)$, respectively. We use $e(G)$ to denote $|E(G)|$. In a *2-cell embedding*, also called *map*, each face is homeomorphic to a disk. The degree $d(v)$ of a vertex v is the number of edges incident with v , counted with multiplicity (loops are counted twice). An edge of a map has two ends (also called *half-edges*), and either one or two sides, depending on the number of faces which is incident with. A map is *rooted* if an edge and one of its half-edges and sides are distinguished as the root-edge, root-end and root-side, respectively. Notice that the rooting of maps on orientable surfaces usually omits the choice of a root-side because the underlying surface is oriented and maps are considered up to orientation preserving homeomorphism. Our choice of a root-side is equivalent in the orientable case to the choice of an orientation of the surface. The root-end and -sides define the root-vertex and -face, respectively. Rooted maps are considered up to cell-preserving homeomorphisms preserving the root-edge, -end, and -side. In figures, the root-edge is indicated as an oriented edge pointing away from the root-end and crossed by an arrow pointing towards the root-side (this last, provides the orientation in the surface).

For a graph G , the *Euler genus* of G , denoted $\gamma(G)$, is the smallest Euler genus among all surfaces in which G can be embedded. Determining the Euler genus of a graph is an NP-hard problem [30], hence we assume throughout the paper that we are given an already embedded graph. An *O-arc* is a subset of Σ homeomorphic to \mathbb{S}^1 . A subset of Σ meeting the drawing only at vertices of G is called *G-normal*. If an *O-arc* is *G-normal*, then we call it a *noose*. The *length* of a noose is the number of its vertices. Many results in topological graph theory rely on the concept of *representativity* [26, 28], also called *face-width*, which is a parameter that quantifies local planarity and density of embeddings. The representativity $\mathbf{rep}(G, \tau)$ of a graph embedding (G, τ) is the smallest length of a non-contractible (i.e., non null-homotopic) noose in Σ . We call an embedding (G, τ) *polyhedral* [24] if G is 3-connected and $\mathbf{rep}(G, \tau) \geq 3$, or if G is a clique and $1 \leq |V(G)| \leq 3$. With abuse of notation, we also say in that case that the graph G itself is polyhedral.

For a given embedding (G, τ) , we denote by (G^*, τ) its dual embedding. Thus G^* is the geometric dual of G . Each vertex v (resp. face r) in (G, τ) corresponds to some face v^* (resp. vertex r^*) in (G^*, τ) . Also, given a set $S \subseteq E(G)$, we denote as S^* the set of the duals of the edges in S . Let (G, τ)

be an embedding and let (G^*, τ) be its dual. We define the *radial graph embedding* (R_G, τ) of (G, τ) (also known as *vertex-face graph embedding*) as follows : R_G is an embedded bipartite graph with vertex set $V(R_G) = V(G) \cup V(G^*)$. For each pair $e = \{v, u\}$, $e^* = \{u^*, v^*\}$ of dual edges in G and G^* , R_G contains edges $\{v, v^*\}$, $\{v^*, u\}$, $\{u, u^*\}$, and $\{u^*, v\}$. Mohar and Thomassen [24] proved that, if $|V(G)| \geq 4$, the following conditions are equivalent : (i) (G, τ) is a polyhedral embedding ; (ii) (G^*, τ) is a polyhedral embedding ; and (iii) (R_G, τ) has no multiple edges and every 4-cycle of R_G is the border of some face. The *medial graph embedding* (M_G, τ) of (G, τ) is the dual embedding of the radial embedding (R_G, τ) of (G, τ) . Note that (M_G, τ) is a Σ -embedded 4-regular graph.

2.3 Tree-like decompositions of graphs

Let G be a graph on n vertices. A *branch decomposition* (T, μ) of a graph G consists of an unrooted ternary tree T (i.e., all internal vertices are of degree three) and a bijection $\mu : L \rightarrow E(G)$ from the set L of leaves of T to the edge set of G . We define for every edge e of T the *middle set* $\mathbf{mid}(e) \subseteq V(G)$ as follows : Let T_1 and T_2 be the two connected components of $T \setminus \{e\}$. Then let G_i be the graph induced by the edge set $\{\mu(f) : f \in L \cap V(T_i)\}$ for $i \in \{1, 2\}$. The *middle set* is the intersection of the vertex sets of G_1 and G_2 , i.e., $\mathbf{mid}(e) := V(G_1) \cap V(G_2)$. The *width* of (T, μ) is the maximum order of the middle sets over all edges of T , i.e., $\mathbf{w}(T, \mu) := \max\{|\mathbf{mid}(e)| : e \in T\}$. An optimal branch decomposition of G is defined by a tree T and a bijection μ which give the minimum width, the *branchwidth*, denoted by $\mathbf{bw}(G)$.

Let $G = (V, E)$ be a connected graph. For $S \subseteq V$, we denote by $\delta(S)$ the set of all edges with an end in S and an end in $V \setminus S$. Let $\{V_1, V_2\}$ be a partition of V . If $G[V \setminus V_1]$ and $G[V \setminus V_2]$ are both non-null and connected, we call $\delta(V_1)$ a *bond* of G [28].

A *carving decomposition* (T, μ) is similar to a branch decomposition, only with the difference that μ is a bijection between the leaves of the tree and the vertex set of the graph G . For an edge e of T , the counterpart of the middle set, called the *cut set* $\mathbf{cut}(e)$, contains the edges of G with endvertices in the leaves of both subtrees. The counterpart of branchwidth is *carvingwidth*, and is denoted by $\mathbf{cw}(G)$. In a *bond carving decomposition*, every cut set is a bond of the graph. That is, in a bond carving decomposition, every cut set separates the graph into two connected components.

Let G_1 and G_2 be graphs with disjoint vertex-sets and let $k \geq 0$ be an integer. For $i = 1, 2$, let $W_i \subseteq V(G_i)$ form a clique of size k and let G'_i ($i = 1, 2$) be obtained from G_i by deleting some (possibly no) edges from $G_i[W_i]$ with both endpoints in W_i . Consider a bijection $h : W_1 \rightarrow W_2$. We define a *clique sum* G of G_1 and G_2 , denoted by $G = G_1 \oplus_k G_2$, to be the graph obtained from the union of G'_1 and G'_2 by identifying w with $h(w)$ for all $w \in W_1$. The integer k is called the *size* of the clique sum.

2.4 The symbolic method and analytic combinatorics

The main reference in enumerative combinatorics is [16]. The framework introduced in this book gives a language to translate combinatorial conditions between combinatorial classes into equations. This is what is called the *symbolic method* in combinatorics. Later, we can treat these equations as relations between analytic functions. This point of view gives us the possibility to use complex analysis techniques to obtain information about the combinatorial classes. This is the origin of the term *analytic combinatorics*.

For a set \mathcal{A} of objects, let $|\cdot|$ be an application from \mathcal{A} to \mathbb{N} , which is called the *size*. A pair $(\mathcal{A}, |\cdot|)$ is called a *combinatorial class*. Define the formal power series (called the *generating function* or *GF* associated to the class) $\mathbf{A}(z) = \sum_{a \in \mathcal{A}} z^{|a|} = \sum_{k=0}^{\infty} a_k z^k$. The constructions we use in this work and their translation into the language of GFs are shown in Table 1.

Construction		GF	
Union	$\mathcal{A} \cup \mathcal{B}$	$\mathbf{A}(z) + \mathbf{B}(z)$	The union $\mathcal{A} \cup \mathcal{B}$ of \mathcal{A} and \mathcal{B} refers to the disjoint union of the classes. The cartesian product $\mathcal{A} \times \mathcal{B}$ of \mathcal{A} and \mathcal{B} is the set $\{(a, b) : a \in \mathcal{A}, b \in \mathcal{B}\}$. The sequence $\text{Seq}(\mathcal{A})$ of a set \mathcal{A} corresponds to the set $\mathcal{E} \cup \mathcal{A} \cup \mathcal{A} \times \mathcal{A} \cup \mathcal{A} \times \mathcal{A} \times \mathcal{A} \cup \dots$. At last, the pointing operator \mathcal{A}^\bullet of a set \mathcal{A} consists in pointing one of the atoms of each element $a \in \mathcal{A}$.
Product	$\mathcal{A} \times \mathcal{B}$	$\mathbf{A}(z)\mathbf{B}(z)$	
Sequence	$\text{Seq}(\mathcal{A})$	$\frac{1}{1-\mathbf{A}(z)}$	
Pointing	\mathcal{A}^\bullet	$z \frac{\partial}{\partial z} \mathbf{A}(z)$	

TABLE 1 – Constructions and translations into GF.

The study of the growth of the coefficients of GFs can be obtained by considering GFs as complex functions which are analytic around $z = 0$. This is the philosophy of analytic combinatorics. The growth behavior of the coefficients depends only on the smallest positive singularity of the GF. Its *location* provides the *exponential growth* of the coefficients, and its *behavior* gives the *subexponential growth* of the coefficients.

The basic results in this area are the so-called *Transfer Theorems for singularity analysis*. These results allows us to deduce asymptotic estimates of an analytic function using its asymptotic expansion near its dominant singularity. The precise statement is claimed in [16]. We use the following reduced version of the theorem (without taking care of technical conditions, which are satisfied in all cases) : let $\mathbf{F}(z)$ be a GF with positive coefficients, such that ρ is its smallest real singularity. Suppose that $\mathbf{F}(z)$ admits a singular expansion around $z = \rho$ of the form $\mathbf{F}(z) \underset{z \rightarrow \rho}{=} C(1 - z/\rho)^{-\alpha} + O\left((1 - z/\rho)^{-\alpha+1/2}\right)$, where C is a constant. Then the Transfer Theorem for singularity analysis states that, for k big enough, the following estimate holds

$$[z^k]\mathbf{F}(z) \underset{k \rightarrow \infty}{=} C \frac{k^{\alpha-1}}{\Gamma(\alpha)} \rho^{-k} \left(1 + O\left(k^{-1/2}\right)\right). \tag{1}$$

3 Examples of Dynamic Programming Algorithms

In this section we present two examples of typical dynamic programming algorithms on graphs of bounded branchwidth. The first algorithm solves the VERTEX COVER problem and belongs in Category (A) while the second solves the CONNECTED VERTEX COVER problem and belongs in Category (C) but not in (B) (nor in (A)).

The standard dynamic programming approach on branch decompositions requires the so called *rooted* branch decomposition defined as a triple (T, μ, e_r) where (T, μ) is a branch-decomposition of G where T is a tree rooted on a leaf v_l incident to some edge e_r of T . We insist that no edge of G is assigned to v_l and thus $\text{mid}(e_r) = \emptyset$ (for this, we take any edge of a branch decomposition, subdivide it and then connect by e_r the subdivision vertex with a new leaf t_l). The edges of T can be oriented towards the root e_r and for each edge $e \in E(T)$ we denote by E_e the edges of G that are mapped to leaves of T that are descendants of e . We also set $G_e = G[E_e]$ and we denote by $L(T)$ the edges of T that are incident to leaves of T . Given an edge e heading at a non-leaf vertex v , we denote by

$e_1, e_2 \in E(T)$ the two edges with tail v . As both examples below are variants of the vertex cover problem, we can also assume that $|E(T)| = O(k \cdot n)$ as, otherwise, the answer for each problem is trivially negative.

Dynamic programming for VERTEX COVER. Let G be a graph and $X, X' \subseteq V(G)$ where $X \cap X' = \emptyset$. We say that $\mathbf{vc}(G, X, X') \leq k$ if G contains a vertex cover S where $|S| \leq k$ and $X \subseteq S \subseteq V(G) \setminus X'$. Let $\mathcal{R}_e = \{(X, k) \mid \mathbf{vc}(G_e, X, \mathbf{mid}(e) \setminus X) \leq k\}$ and observe that $\mathbf{vc}(G) \leq k$ iff $(\emptyset, k) \in \mathcal{R}_e$. For each $e \in E(T)$ we can compute \mathcal{R}_e by using the following dynamic programming formula :

$$\mathcal{R}_e = \begin{cases} \{(X, k) \mid X \neq \emptyset \wedge k \geq |X|\} & \text{if } e \in L(T) \\ \{(X, k) \mid \exists (X_1, k_1) \in \mathcal{R}_{e_1}, \exists (X_2, k_2) \in \mathcal{R}_{e_2} : \\ (X_1 \cup X_2) \cap \mathbf{mid}(e) = X \wedge k_1 + k_2 - |X_1 \cap X_2| \leq k\} & \text{if } e \notin L(T) \end{cases}$$

Notice that for each $e \in E(T)$, $|\mathcal{R}_e| \leq 2^{|\mathbf{mid}(e)|} \cdot k$. Therefore, the above algorithm can check whether $\mathbf{vc}(G) \leq k$ in $O(2^{\mathbf{bw}(G)} \cdot k \cdot |V(T)|)$ steps. Clearly, this simple algorithm is single exponential on $\mathbf{bw}(G)$. Moreover the above dynamic programming machinery can be adapted to many other combinatorial problems where the certificate of the solution is a (non-restricted) subset of vertices (e.g. DOMINATING SET, 3-COLORING, INDEPENDENT SET, among others).

Dynamic programming for CONNECTED VERTEX COVER. Suppose now that we are looking for a *connected* vertex cover of size $\leq k$. Clearly, the above dynamic programming formula does not work for this variant as we should book-keep more information on X towards encoding the connectivity demand.

Let G be a graph, $X \subseteq V(G)$ and \mathcal{H} be a (possibly empty) hypergraph whose vertex set is a subset of X , whose hyperedges are non-empty, pairwise non-intersecting, and such that each vertex of \mathcal{H} belongs to some of its hyperedges (we call such a hypergraph *partial packing* of X). Suppose that \mathcal{H} is a partial packing on $\mathbf{mid}(e)$. We say that $\mathbf{cvc}(G, \mathcal{H}) \leq k$ if G contains a vertex cover S where $|S| \leq k$ and such that if \mathcal{C} is the collection of the connected components of $G_e[S]$, then either $|E(\mathcal{H})| = |\mathcal{C}|$ and $(X, \{X \cap V(C) \mid C \in \mathcal{C}\}) = \mathcal{H}$ or $E(\mathcal{H}) = \emptyset$ and $|\mathcal{C}| = 1$.

As before, let $\mathcal{Q}_e = \{(\mathcal{H}, k) \mid \mathbf{cvc}(G, \mathcal{H}) \leq k\}$ and observe that $\mathbf{cvc}(G) \leq k$ iff $(\emptyset, k) \in \mathcal{Q}_e$. The dynamic programming formula for computing \mathcal{Q}_e for each $e \in E(T)$ is the following.

$$\mathcal{Q}_e = \begin{cases} \{(\mathcal{H}, k) \mid \min\{k, |E(\mathcal{H})| + 1\} \geq |V(\mathcal{H})| \geq 1\} & \text{if } e \in L(T) \\ \{(\mathcal{H}, k) \mid \exists (\mathcal{H}_1, k_1) \in \mathcal{Q}_{e_1}, \exists (\mathcal{H}_2, k_2) \in \mathcal{Q}_{e_2} : \\ V(\mathcal{H}_1) \cap (\mathbf{mid}(e_1) \cap \mathbf{mid}(e_2)) = V(\mathcal{H}_2) \cap (\mathbf{mid}(e_1) \cap \mathbf{mid}(e_2)), \\ (\mathcal{H}_1 \oplus \mathcal{H}_2)[\mathbf{mid}(e)] = \mathcal{H}, k_1 + k_2 - |V(\mathcal{H}_1) \cap V(\mathcal{H}_2)| \leq k, \\ \text{if } E(\mathcal{H}) = \emptyset \text{ then } |E(\mathcal{H}_1 \oplus \mathcal{H}_2)| = 1, \text{ and} \\ \text{if } E(\mathcal{H}) \neq \emptyset \text{ then } |E(\mathcal{H}_1 \oplus \mathcal{H}_2)| = |E(\mathcal{H})|\} & \text{if } e \notin L(T). \end{cases}$$

In the above formula, $\mathcal{H}_1 \oplus \mathcal{H}_2$ is the hypergraph with vertex set $V(\mathcal{H}_1) \cup V(\mathcal{H}_2)$ where each of its hyperedges contains the vertices of each of the connected components of $\mathcal{H}_1 \cup \mathcal{H}_2$.

Clearly each \mathcal{H} corresponds to a collection of subsets of X and the number of such collections for a given set $\mathbf{mid}(e)$ of r elements is given by the r -th Bell number of r , denoted by B_r . By taking the straightforward upper bound $|B_r| = 2^{O(r \log r)}$, we have that one can check whether an input graph G has a connected vertex cover of size at most k in $O(2^{\mathbf{bw}(G) \cdot \log(\mathbf{bw}(G))} \cdot k \cdot |V(T)|)$ steps.

As the growth of B_r is not single exponential, we cannot hope for a single exponential (in $\mathbf{bw}(G)$) running time for the above dynamic programming procedure and no algorithm is known for this problem that runs in time that is single exponential in $\mathbf{bw}(G)$. The same problem appears for numerous other problems where further restrictions apply to their solution certificates. Such problems can be connected variants of problems in Category (A) and others such as MAXIMUM INDUCED FOREST, MAXIMUM d -DEGREE-BOUNDED CONNECTED SUBGRAPH, MAXIMUM d -DEGREE-BOUNDED CONNECTED INDUCED SUBGRAPH and all the variants studied in [27], CONNECTED DOMINATING SET, CONNECTED r -DOMINATION, CONNECTED FVS, MAXIMUM LEAF SPANNING TREE, MAXIMUM FULL-DEGREE SPANNING TREE, MAXIMUM EULERIAN SUBGRAPH, STEINER TREE, or MAXIMUM LEAF TREE.

4 Polyhedral Decompositions

We introduce in this section *polyhedral decompositions* of graphs embedded in surfaces. Let G be an embedded graph, and let N be a noose in the surface. Similarly to [6], we use the notation $G \gg N$ for the graph obtained by cutting G along the noose N and gluing a disk on the obtained boundaries.

Definition 4.1 *Given a graph $G = (V, E)$ embedded in a surface of Euler genus γ , a polyhedral decomposition of G is a set of graphs $\mathcal{G} = \{H_1, \dots, H_\ell\}$ together with a set of vertices $A \subseteq V$ such that*

- $|A| = O(\gamma)$;
- H_i is a minor of $G[V \setminus A]$, for $i = 1, \dots, \ell$;
- H_i has a polyhedral embedding in a surface of Euler genus at most γ , for $i = 1, \dots, \ell$;
- $G[V \setminus A]$ can be constructed by joining the graphs of \mathcal{G} applying clique sums of size 0, 1, or 2.

Remark 4.1 *Note that an embedded graph H is not polyhedral if and only if there exists a noose N of length at most 2 in the surface in which H is embedded, such that either N is non-contractible or $V(H) \cap N$ separates H . Indeed, if H has representativity at most 2, then there exists a non-contractible noose N of length at most 2. Otherwise, since H is not polyhedral, H has a minimal separator S of size at most 2. It is then easy to see that there exists a noose containing only vertices of S .*

Algorithm 1 provides an efficient way to construct a polyhedral decomposition, as it is stated in Proposition 4.1.

In the above algorithm, the addition of an edge $\{u, v\}$ represents the existence of a path in G between u and v that is not contained in the current component.

Proposition 4.1 *Given a graph G on n vertices embedded in a surface, Algorithm 1 constructs a polyhedral decomposition of G in $O(n^3)$ steps.*

Algorithm 1 Construction of a polyhedral decomposition of an embedded graph G **Input:** A graph G embedded in a surface of Euler genus γ .**Output:** A polyhedral decomposition of G . $A = \emptyset$, $\mathcal{G} = \{G\}$ (the elements in \mathcal{G} , which are embedded graphs, are called *components*).**while** \mathcal{G} contains a non-polyhedral component H **do**Let N be a noose as described in Observation 4.1 in the surface in which H is embedded,
and let $S = V(H) \cap N$.**if** N is non-surface-separating **then**Add S to A , and replace in \mathcal{G} component H with $H[V(H) \setminus S]$.**if** N is surface-separating **then**Let H_1, H_2 be the subgraphs of H corresponding to the two surfaces occurring after splitting H **if** $S = \{u\} \cup \{v\}$ and $\{u, v\} \notin E(H)$ **then**Add the edge $\{u, v\}$ to H_i , $i = 1, 2$.Replace in \mathcal{G} component H with the components of H containing at least one edge of H .**return** $\{\mathcal{G}, A\}$.**Proof:** We first prove that the output $\{\mathcal{G}, A\}$ of Algorithm 1 is indeed a polyhedral decomposition of G , and then we analyze the running time.Let us see that each component of \mathcal{G} is a minor of $G[V \setminus A]$. Indeed, the only edges added to G by Algorithm 1 are those between two non-adjacent vertices u, v that separate a component H into several components H_1, \dots, H_ℓ . For each component H_i , $i = 1, \dots, \ell$, there exists a path between u and v in $H \setminus H_i$ (provided that the separators of size 1 have been already removed, which can we assumed w.l.o.g.), and therefore the graph obtained from H_i by adding the edge $\{u, v\}$ is a minor of H , which is inductively a minor of $G[V \setminus A]$. Also, each component of \mathcal{G} is polyhedral by definition of the algorithm.As a non-separating noose is necessarily non-contractible, each time some vertices are moved to A the Euler genus of the surfaces strictly decreases [24, Lemma 4.2.4]. Therefore, $|A| = O(\gamma)$.By the construction of the algorithm, it is also clear that each component of \mathcal{G} has a polyhedral embedding in a surface of Euler genus at most γ . Finally, $G[V \setminus A]$ can be constructed by joining the graphs of \mathcal{G} applying clique sums of size 0, 1, or 2.Thus, $\{\mathcal{G}, A\}$ is a polyhedral decomposition of G by Definition 4.1.We now analyze the running time of the algorithm. Separators of size at most 2 can be found in $O(n^2)$ steps [22]. A noose with respect to a graph H corresponds to a cycle in the radial graph of H , hence can also be found³ in $O(n^2)$ (using that the number of edges of a bounded-genus graph is linearly bounded by its number of vertices). Since each time that we find a *small* separator we decrease the size of the components, the running time of the algorithm is $O(n^3)$. \square ³A shortest non-contractible cycle can be found in $O(2^{O(\gamma \log \gamma)} n^{4/3})$ steps [6]. This time improves on $O(n^3)$ for a big range of values of γ .

5 Some Topological Lemmata

In this section we state some topological lemmata that are used in Sections 6 and 7. In particular, Lemmata 5.1 and 5.2 are used in the proof of Theorem 6.1 while Lemma 5.3 is used in the proof of Theorem 7.1.

Given a graph G embedded in a surface of Euler genus γ , its dual G^* and a spanning tree C^* of G^* , we call $C = \{e \in E(G) \mid e^* \in E(C^*)\}$ a *spanning cotree* of G . We define a *tree-cotree partition* (cf. [15]) of an embedded graph G to be a triple (T, C, X) where T is a spanning tree of G , C is a spanning cotree of G , $X \subseteq E(G)$, and the three sets $E(T)$, C , and X form a partition of $E(G)$. Eppstein proved [15, Lemma 3.1] that if T and C^* are forests such that $E(T)$ and C are disjoint, we can make T become part of a spanning tree T' and C become part of a spanning cotree disjoint from T' , extending T and C to a tree-cotree decomposition. We can now announce the following lemma from [15, Lemma 3.2].

Lemma 5.1 (Eppstein [15]) *If (T, C, X) is a tree-cotree decomposition of a graph G embedded in a surface of Euler genus γ , then $|X| = O(\gamma)$.*

Lemma 5.2 *Let Σ be a surface without boundary. Let S be a set of (not necessary disjoint) $O(\gamma(\Sigma))$ cycles such that $\Sigma \setminus S$ has 2 connected components. Let H be the graph corresponding to the union of the cycles in S . Then $\sum_{v \in V(H)} (d(v) - 2) = O(\gamma(\Sigma))$.*

Proof: Let κ be the number of cycles, so by assumption $\kappa = O(\gamma(\Sigma))$. The first step consists in simplifying the problem. Let H' be the graph obtained from H by dissolving vertices of degree 2 (except from the case of isolated cycles, where we obtain a single vertex of degree 2 and a loop). Each dissolved vertex had degree 2, so the sum $\sum_{v \in V(H')} (d(v) - 2)$ coincides with $\sum_{v \in V(H)} (d(v) - 2)$. Additionally, by assumption H' separates Σ into 2 connected components Σ' and Σ'' . Let H'_1, H'_2, \dots, H'_r be the maximal connected subgraphs of H' . In particular, $r \leq \kappa$.

Some of these connected subgraphs may be incident with Σ' but not with Σ'' , or conversely. Additionally, there is at least one connected subgraph H'_i incident with both connected components. W.l.o.g. we assume that the subgraphs H'_1, H'_2, \dots, H'_p are incident only with Σ' , H'_{p+1}, \dots, H'_q are incident with both components, and H'_{q+1}, \dots, H'_r are incident only with Σ'' . It is obvious that there exists a path joining a vertex of H'_i with a vertex of H'_{i+1} if $1 \leq i \leq q - 1$ or $p + 1 \leq i \leq r - 1$.

From graphs $H'_1, H'_2, \dots, H'_p, \dots, H'_q$ (the ones which are incident with Σ') we construct a new graph G_1 in the following inductive way : we start taking H'_q and H'_{q-1} , and a path joining a vertex of H'_q to a vertex of H'_{q-1} . This path exists because H'_q and H'_{q-1} are incident with Σ' . Consider the graph obtained from H'_q and H'_{q-1} by adding an edge that joins this pair of vertices. Then, delete H'_q and H'_{q-1} from the initial list and add this new connected graph. This procedure is done $q - 1$ times. At the end, we obtain a connected graph G' incident with both Σ' and Σ'' where each vertex has degree at least 3. Finally, we apply the same procedure with $G', H'_{q+1}, \dots, H'_r$, obtaining a connected graph G . Observe also that

$$\sum_{v \in V(H)} (d(v) - 2) \leq \sum_{v \in V(G)} (d(v) - 2) < \sum_{v \in V(G)} d(v) = 2|E(G)|.$$

In what follows, we obtain upper bounds for $2|E(G)|$. Observe that H' defines a pair of faces over Σ , not necessarily disks. In the previous construction of G , every time we add an edge either we subdivide a face into two parts or not. Consequently, the number of faces that G defines over Σ is at most $2 + \kappa$. The next step consists in reducing the surface in the following way : let f be a face determined by G over Σ . If f is contractible, we do nothing. If not, there is a non-contractible cycle \mathbb{S}^1 contained on f . Consider the connected component of $\Sigma \succ \mathbb{S}^1$ which contains G (call it Σ_1). It is obvious that G defines a decomposition of Σ_1 , $\gamma(\Sigma_1) \leq \gamma(\Sigma)$, and the number of faces has been increased by at most one. Observe that for each operation \succ we reduce the Euler genus and we create at most one face. As the Euler genus is finite, the number of \succ operations is also finite. This gives rise to a surface Σ_s , where $s \leq \gamma(\Sigma)$, such that all faces determined by G are contractible. Additionally, the number of faces that G determines over Σ_s is smaller than $2 + \kappa + \gamma(\Sigma)$.

G defines a map on Σ_s (i.e., all faces are contractible), and consequently we can apply Euler's formula. Then $|F(G)| + |V(G)| = |E(G)| + 2 - \gamma(\Sigma_s)$. Then, $|F(G)| \leq 2 + \kappa + \gamma(\Sigma)$, so $|E(G)| + 2 - \gamma(\Sigma_s) = |V(G)| + |F(G)| \leq |V(G)| + 2 + \kappa + \gamma(\Sigma)$. The degree of each vertex is at least 3, thus $3|V(G)| \leq 2|E(G)|$. Substituting this condition in the previous equation, we obtain

$$|E(G)| + 2 - \gamma(\Sigma_s) \leq |V(G)| + 2 + \kappa + \gamma(\Sigma) \leq \frac{2}{3}|E(G)| + 2 + \kappa + \gamma(\Sigma).$$

Isolating $|E(G)|$, we get that $2|E(G)| \leq 6\kappa + 6\gamma(\Sigma_s) + 6\gamma(\Sigma) \leq 6\kappa + 12\gamma(\Sigma)$. This bound immediately translates into the statement of the Lemma. \square

We need another topological result, which deals with cycles that separate a given surface, and whose proof is an immediate consequence of [24, Proposition 4.2.1].

Lemma 5.3 *Let Σ be a surface with boundary and let \mathbb{S}^1 be a separating cycle. Let V_1 and V_2 be the connected components of $\Sigma \succ \mathbb{S}^1$. Then $\gamma(\Sigma) = \gamma(V_1) + \gamma(V_2)$.*

6 Surface Cut Decompositions

Sphere cut decompositions [28] have proved to be very useful to analyze the running time of algorithms based on dynamic programming over branch decompositions on planar graphs [12–14, 27]. In this section we generalize sphere cut decompositions to graphs on surfaces ; we call them *surface cut decompositions*. First we need a topological definition. A subset Π of a surface Σ is *fat-connected* if for every two points $p, q \in \Pi$, there exists a path $P \subseteq \Pi$ such that for every $x \in P$, $x \neq p, q$, there exists a subset D homeomorphic to an open disk such that $x \in D \subseteq \Pi$. We can now define the notion of surface cut decomposition.

Definition 6.1 *Given a graph G embedded in a surface Σ , a surface cut decomposition of G is a branch decomposition (T, μ) of G such that, for each edge $e \in E(T)$, there is a subset of vertices $A_e \subseteq V(G)$ with $|A_e| = O(\gamma(\Sigma))$ and either*

- $|\text{mid}(e) \setminus A_e| \leq 2$, or
- there exists a polyhedral decomposition $\{\mathcal{G}, \mathcal{A}\}$ of G and a graph $H \in \mathcal{G}$ such that

- $A_e \subseteq A$;
- $\mathbf{mid}(e) \setminus A_e \subseteq V(H)$;
- the vertices in $\mathbf{mid}(e) \setminus A_e$ are contained in a set N of $O(\gamma(\Sigma))$ nooses, such that the total number of occurrences in N of the vertices in $\mathbf{mid}(e) \setminus A_e$ is $|\mathbf{mid}(e) \setminus A_e| + O(\gamma(\Sigma))$; and
- $\Sigma \setminus \bigcup_{N \in \mathcal{N}} N$ contains exactly two connected components, which are both fat-connected.

Note that a sphere cut decomposition is a particular case of a surface cut decomposition when $\gamma = 0$, by taking $A_e = \emptyset$, \mathcal{G} containing only the graph itself, and all the vertices of each middle set contained in a single noose.

We need some definitions and auxiliary results to be applied for building surface cut decompositions. In the same spirit of [20, Theorem 1] we can prove the following lemma. We omit the proof here since the details are very similar⁴ to the proof in [20].

Lemma 6.1 *Let (G, τ) and (G^*, τ) be dual polyhedral embeddings in a surface of Euler genus γ and let (M_G, τ) be the medial graph embedding. Then $\max\{\mathbf{bw}(G), \mathbf{bw}(G^*)\} \leq \mathbf{cw}(M_G)/2 \leq 6 \cdot \mathbf{bw}(G) + 2\gamma + O(1)$. In addition, given a branch decomposition of G of width at most k , a carving decomposition of M_G of width at most $12k$ can be found in linear time.*

Lemma 6.2 (folklore) *The removal of a vertex from a non-acyclic graph decreases its branchwidth by at most 1.*

Lemma 6.3 *Let G_1 and G_2 be graphs with at most one vertex in common. Then $\mathbf{bw}(G_1 \cup G_2) = \max\{\mathbf{bw}(G_1), \mathbf{bw}(G_2)\}$.*

Proof: Assume first that G_1 and G_2 share one vertex v . Clearly $\mathbf{bw}(G_1 \cup G_2) \geq \max\{\mathbf{bw}(G_1), \mathbf{bw}(G_2)\}$. Conversely, for $i = 1, 2$, let (T_i, μ_i) be a branch decomposition of G_i such that $\mathbf{w}(T_i, \mu_i) \leq k$. For $i = 1, 2$, let T_i^v be the minimal subtree of T_i containing all the leaves u_i of T_i such that v is an endpoint of $\mu_i(u_i)$. For $i = 1, 2$, we take an arbitrary edge $\{a_i, b_i\}$ of T_i^v , we subdivide it by adding a new vertex w_i , and then we build a tree T from T_1 and T_2 by adding the edge $\{w_1, w_2\}$. We claim that $(T, \mu_1 \cup \mu_2)$ is a branch decomposition of $G_1 \cup G_2$ of width at most k . Indeed, let us compare the middle sets of $(T, \mu_1 \cup \mu_2)$ to those of (T_1, μ_1) and (T_2, μ_2) . First, it is clear that the vertices of $V(G_1) \cup V(G_2) - \{v\}$ appear in $(T, \mu_1 \cup \mu_2)$ in the same middle sets as in (T_1, μ_1) and (T_2, μ_2) . Secondly, $\mathbf{mid}(\{w_1, w_2\}) = \{v\}$, since v is a cut-vertex of $G_1 \cup G_2$. Also, for $i = 1, 2$, $\mathbf{mid}(\{a_i, w_i\}) = \mathbf{mid}(\{w_i, b_i\}) = \mathbf{mid}(\{a_i, b_i\})$, and the latter has size at most k as $\mathbf{w}(T_i, \mu_i) \leq k$. For all other edges e of T_i , $i = 1, 2$, $\mathbf{mid}(e)$ is exactly the same in T and in T_i , since if $e \in E(T_i^v)$ then $v \in \mathbf{mid}(e)$ in both T and T_i , and if $e \in E(T_i \setminus T_i^v)$ then $v \notin \mathbf{mid}(e)$ in both T and T_i .

If G_1 and G_2 share no vertices, we can merge two branch decompositions (T_1, μ_1) and (T_2, μ_2) by subdividing a pair of arbitrary edges, without increasing the width. \square

⁴The improvement in the multiplicative factor of the Euler genus is obtained by applying more carefully Euler's formula in the proof analogous to that of [20, Lemma 2].

Lemma 6.4 ([19]) *Let G_1 and G_2 be graphs with one edge f in common. Then $\mathbf{bw}(G_1 \cup G_2) \leq \max\{\mathbf{bw}(G_1), \mathbf{bw}(G_2), 2\}$. Moreover, if both endpoints of f have degree at least 2 in at least one of the graphs, then $\mathbf{bw}(G_1 \cup G_2) = \max\{\mathbf{bw}(G_1), \mathbf{bw}(G_2)\}$.*

Lemma 6.5 *Let G be a graph and let \mathcal{G} be a collection of graphs such that G can be constructed by joining graphs in \mathcal{G} applying clique sums of size 0, 1, or 2. Given branch decompositions $\{(T_H, \mu_H) \mid H \in \mathcal{G}\}$, we can compute in linear time a branch decomposition (T, μ) of G such that $\mathbf{w}(T, \mu) \leq \max\{2, \{\mathbf{w}(T_H, \mu_H) \mid H \in \mathcal{G}\}\}$. In particular, $\mathbf{bw}(G) \leq \max\{2, \{\mathbf{bw}(H) \mid H \in \mathcal{G}\}\}$.*

Proof: Note that if G_1 and G_2 are graphs with no vertex (resp. a vertex, an edge) in common, then $G_1 \cup G_2 = G_1 \oplus_0 G_2$ (resp. $G_1 \oplus_1 G_2, G_1 \oplus_2 G_2$). We have to show how to merge branch decompositions $(T_1, \mu_1), (T_2, \mu_2)$ of two graphs H_1, H_2 in \mathcal{G} . We distinguish four cases :

- (a) H_1 and H_2 share two vertices v_1, v_2 , and the edge $e = \{v_1, v_2\} \in E(G)$. We take the leaves in T_1 and T_2 corresponding to e , we identify them, and we add a new edge whose leaf corresponds to e (see Figure 1(a)).
- (b) H_1 and H_2 share two vertices v_1, v_2 , and the edge $e = \{v_1, v_2\} \notin E(G)$. We take the leaves in T_1 and T_2 corresponding to e , we identify them, and we dissolve the common vertex (see Figure 1(b)).
- (c) H_1 and H_2 share one vertex v . We take two edges b, c in T_1, T_2 whose leaves correspond to edges containing v , we subdivide them and add a new edge between the newly created vertices (see Figure 1(c)).
- (d) H_1 and H_2 share no vertices. We do the construction of case (c) for any two edges of the two branch decompositions.

The above construction does not increase the branchwidth by Lemmata 6.3 and 6.4. □

Given an embedded graph G and a carving decomposition (T, μ) of its medial graph M_G , we define a *radial decomposition* (T^*, μ^*) of the dual graph R_G , where $T^* = T$ and μ^* is a bijection from the leaves of T to the set of faces of R_G defined as follows : for each edge $e \in E(T)$, $\mu^*(e) = f$, where f is the face in R_G corresponding to the vertex $u_f \in V(M_G)$ such that $\mu(e) = u_f$. Each edge $e \in E(T^*)$ partitions the faces of R_G into two sets F_1 and F_2 . We define the *border set* of e , denoted $\mathbf{bor}(e)$, as the set of edges of R_G that belong to both F_1 and F_2 . Note that F_1 and F_2 may intersect also in vertices, not only in edges. If (T, μ) is a bond carving decomposition of M_G , then the associated radial decomposition (also called *bond*) has nice connectivity properties. Indeed, in a bond carving decomposition, every cut set partitions the vertices of M_G into two subsets V_1, V_2 such that both $M_G[V_1]$ and $M_G[V_2]$ are non-null and connected. This property, seen in the radial decomposition of R_G , implies that each edge $e \in E(T^*)$ corresponds to a partition of the faces of R_G into two sets F_1 and F_2 , namely *black* and *white* faces (naturally partitioning the edges into *black*, *white*, and *grey*), such that it is possible to reach any black (resp. white) face from any black (resp. white) face only crossing black (resp. white) edges. In other words, each set of monochromatic faces is fat-connected.

Remark 6.1 *Recall that all the faces of a radial graph R_G are tiles, that is, each face has exactly 4 edges. Also, each one of those tiles corresponds to a pair of dual edges e and e^* of G and G^* , respectively. Given a carving decomposition (T, μ) of M_G (or equivalently, a radial decomposition*

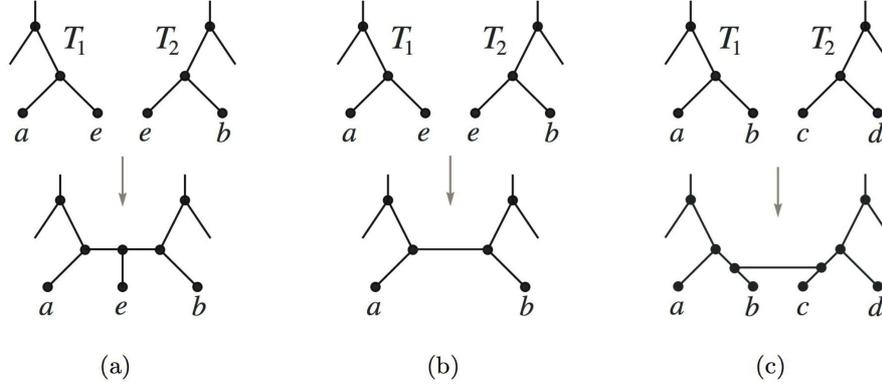


FIG. 1 – Merging branch decompositions (T_1, μ_1) and (T_2, μ_2) of two components H_1 and H_2 in a polyhedral decomposition $\{\mathcal{G}, A\}$ of $G = (V, E)$. There are three cases : (a) H_1 and H_2 share two vertices v_1, v_2 and the edge $e = \{v_1, v_2\}$ is in E ; (b) H_1 and H_2 share two vertices v_1, v_2 and $e = \{v_1, v_2\}$ is *not* in E ; (c) H_1 and H_2 share one vertex v .

(T^*, μ^*) of R_G), one can obtain in a natural way branch decompositions of G and G^* by redefining the bijection μ from the leaves of T to the edges of G (or G^*) that correspond to the faces of R_G .

We provide now an algorithm to construct a surface graph decomposition of an embedded graph. The proof of Theorem 6.1 uses Proposition 4.1, topological Lemmata 5.1 and 5.2, and the results of the current section.

Algorithm 2 Construction of a surface cut decomposition of an embedded graph G

Input: An embedded graph G .

Output: A surface cut decomposition of G .

Compute a polyhedral decomposition $\{\mathcal{G}, A\}$ of G , using Algorithm 1.

for each component H of \mathcal{G} **do**

1. Compute a branch decomposition (T'_H, μ'_H) of H , using [1, Theorem 3.8].

2. Transform (T'_H, μ'_H) to a carving decomposition (T^c_H, μ^c_H) of the medial graph M_H , using Lemma 6.1.

3. Transform (T^c_H, μ^c_H) to a *bond* carving decomposition (T^b_H, μ^b_H) of M_H , using [28].

4. Transform (T^b_H, μ^b_H) to a branch decomposition (T_H, μ_H) of H , using Observation 6.1.

Construct a branch decomposition (T, μ) of G by merging, using Lemma 6.5, the branch decompositions $\{(T_H, \mu_H) \mid H \in \mathcal{G}\}$, and by adding the set of vertices A to all the middle sets.

return (T, μ) .

Theorem 6.1 Given a graph G on n vertices embedded in a surface of Euler genus γ , with $\mathbf{bw}(G) \leq k$, Algorithm 2 constructs, in $2^{3k+O(\log k)}n^3$ steps, a surface cut decomposition (T, μ) of G of width at most $27k + O(\gamma)$.

Proof: We prove, in this order, that (1) the output (T, μ) of Algorithm 2 is indeed a surface cut decomposition of G ; (2) the width of (T, μ) is at most $27\mathbf{bw}(G) + \mathcal{O}(\gamma)$; and (3) the claimed running time.

(1) (T, μ) is a surface cut decomposition of G .

We shall prove that all the properties of Definition 6.1 are fulfilled. For each $e \in E(T)$ we set $A_e = A \cap \mathbf{mid}(e)$, where A is the set of vertices output by Algorithm 1. Hence, by Proposition 4.1, $|A| = \mathcal{O}(\gamma)$.

By construction, it is clear that (T, μ) is a branch decomposition of G . In (T, μ) , there are some edges that have been added in the last step of Algorithm 2, in order to merge branch decompositions of the graphs in \mathcal{G} , with the help of Lemma 6.5. Let e be such an edge. Since $\{\mathcal{G}, A\}$ is a polyhedral decomposition of G , any pair of graphs in \mathcal{G} share at most 2 vertices, hence $|\mathbf{mid}(e) \setminus A_e| \leq 2$.

All other edges of (T, μ) correspond to an edge of a branch decomposition of some polyhedral component $H \in \mathcal{G}$. Let henceforth e be such an edge. Therefore, $\mathbf{mid}(e) \setminus A_e \subseteq V(H)$. To complete this part of the proof, we prove in a sequence of three claims that the remaining conditions of Definition 6.1 hold.

Claim 1 *The vertices in $\mathbf{mid}(e) \setminus A_e$ are contained in a set \mathcal{N} of $\mathcal{O}(\gamma)$ nooses.*

Proof: The proof uses the tree-cotree partition defined in Section 5.

Recall that e is an edge that corresponds to a branch decomposition (T_H, μ_H) of a polyhedral component H of \mathcal{G} . The branch decomposition (T_H, μ_H) of H has been built by Algorithm 2 from a bond carving decomposition of its medial graph M_H , or equivalently from a bond radial decomposition of its radial graph R_H . Due to the fact that the carving decomposition of M_H is bond, edge e partitions the vertices of M_H into two sets – namely, *black* and *white* vertices – each one inducing a connected subgraph of M_H . There are three types of edges: *black*, *white*, and *grey*, according to whether they belong to faces of the same color (black or white) or not. Therefore, the corresponding black and white faces also induce connected subgraphs of R_H , in the sense that it is always possible to reach any black (resp. white) face from any black (resp. white) face only crossing black (resp. white) edges.

Let us now see which is the structure of the subgraph of R_H induced by the edges F belonging to both black and white faces look like. Since each edge of R_H contains a vertex of H and another from H^* , the vertices in $\mathbf{mid}(e)$ are contained in $R_H[F]$, so it suffices to prove that $R_H[F]$ can be partitioned into a set of $\mathcal{O}(\gamma)$ cycles (possibly sharing some vertices).

To this end, first note that in $R_H[F]$ all vertices have even degree. Indeed, let $v \in V(R_H[F])$, and consider a clockwise orientation of the edges incident to v in $R_H[F]$. Each of such edges alternates from a black to a white face, or viceversa, so beginning from an arbitrary edge and visiting all others edges in the clockwise order, we deduce that the number of edges incident to v is necessarily even.

Therefore, $R_H[F]$ can be partitioned into a set of cycles. Let us now bound the number of such cycles. For simplicity, let us identify a pair of dual edges e and e^* as the same edge. Since the subgraph induced by the black (resp. white) faces is connected, we can consider a spanning tree T_B^* (resp. T_W^*) of the black (resp. white) faces. Merge both trees by adding an edge e_0^* , and

let T^* be the resulting tree. Let T be a spanning tree of the dual graph disjoint from T^* (such a spanning tree exists by [15, Lemma 3.1]). Now consider the tree-cotree partition (T, T^*, X) , where X is the set of edges of R_H that are neither in T nor in T^* .

The edges of T^* , except e_0^* , separate faces of the same color. Therefore, the set $F \in E(R_H)$ of edges separating faces of different color is contained in $T \cup \{e_0\} \cup X$. Since T is a tree, each cycle of $R_H[F]$ uses at least one edge in $\{e_0\} \cup X$. Therefore, $R_H[F]$ can be partitioned into at most $1 + |X|$ cycles. The result follows from the fact that (T, T^*, X) is a tree-cotree partition and therefore $|X| = O(\gamma)$ by Lemma 5.1. \square

Claim 2 $\bigcup_{N \in \mathcal{N}} N$ separates Σ into 2 fat-connected components.

Proof: By Claim 1, the vertices in $\mathbf{mid}(e) \setminus A_e$ are contained in $\bigcup_{N \in \mathcal{N}} N$. The claim holds from the fact that for each component H of \mathcal{G} , (T_H^b, μ_H^b) is a bond carving decomposition of M_H , and by taking into account the discussion before Observation 6.1 in Section 6. \square

Claim 3 The total number of occurrences in \mathcal{N} of the vertices in $\mathbf{mid}(e) \setminus A_e$ is $|\mathbf{mid}(e) \setminus A_e| + O(\gamma)$.

Proof: By Claim 2, $\bigcup_{N \in \mathcal{N}} N$ separates Σ into 2 fat-connected components. Let H be the graph induced in Σ by the nooses in \mathcal{N} . The claim can then be rephrased as $\sum_{v \in V(H)} (d(v) - 2) = O(\gamma)$, which holds by Lemma 5.2 in Section 5. \square

(2) The width of (T, μ) is at most $27 \cdot \mathbf{bw}(G) + O(\gamma)$.

For simplicity, let $k = \mathbf{bw}(G)$. By Proposition 4.1, each polyhedral component H is a minor of G , hence $\mathbf{bw}(H) \leq k$ for all $H \in \mathcal{G}$. In Step 1 of Algorithm 2, we compute a branch decomposition (T_H', μ_H') of H of width at most $k' = \frac{9}{2}k$, using Amir's algorithm [1, Theorem 3.8]. In Step 2, we transform (T_H', μ_H') to a carving decomposition (T_H^c, μ_H^c) of the medial graph M_H of H of width at most $12k'$, using Lemma 6.1. In Step 3, we transform (T_H^c, μ_H^c) to a bond carving decomposition (T_H^b, μ_H^b) of M_H of width at most $12k'$, using the algorithm of [28]. Then, using Observation 6.1, we transform in Step 4 (T_H^b, μ_H^b) to a branch decomposition (T_H, μ_H) of H . By the proof of Claim 1, the discrepancy between $\mathbf{w}(T_H, \mu_H)$ and $\mathbf{w}(T_H^b, \mu_H^b)/2$ is at most the bound provided by Lemma 5.2, i.e., $O(\gamma)$. Therefore, $\mathbf{w}(T_H, \mu_H) \leq 6k' + O(\gamma) = 27k + O(\gamma)$, for all $H \in \mathcal{G}$.

Finally, we merge the branch decompositions of the polyhedral components to obtain a branch decomposition (T, μ) of G , by adding the vertices in A to all the middle sets. Combining the discussion above with Lemmata 6.2 and 6.5, and using that $|A| = O(\gamma)$, we get that

$$\begin{aligned} \mathbf{w}(T, \mu) &\leq \max\{2, \{\mathbf{w}(T_H, \mu_H) \mid H \in \mathcal{G}\}\} + |A| \\ &\leq 27k + O(\gamma) + |A| \\ &= 27k + O(\gamma). \end{aligned}$$

(3) Algorithm 2 runs in $2^{3k+O(\log k)} n^3$ steps.

We analyze sequentially the running time of each step. First, we compute a polyhedral decomposition of G using Algorithm 1 in $O(n^3)$ steps, by Proposition 4.1. Then, we run Amir's algorithm in each component in Step 1, which takes $O(2^{3k}k^{3/2}n^2)$ steps [1, Theorem 3.8]. Step 2 can be done in linear time by Lemma 6.1. Step 3 can be done in $O(n^2)$ time [28]. Step 4 takes linear time by Observation 6.1. Merging the branch decompositions can clearly be done in linear time. Finally, since any two elements in \mathcal{G} share at most two vertices, the overall running time is the claimed one. \square

How surface cut decompositions are used for dynamic programming. We shall now discuss how surface cut decompositions guarantee good upper bounds on the size of the tables of dynamic programming algorithms for problems in Category (C). The size of the tables depends on how many ways a partial solution can intersect a middle set during the dynamic programming algorithm. The interest of a surface cut decomposition is that the middle sets are placed on the surface in such a way that permits to give a precise asymptotic enumeration of the size of the tables. Indeed, in a surface cut decomposition, once we remove a set of vertices whose size is linearly bounded by γ , the middle sets are either of size at most two (in which case the size of the tables is bounded by a constant) or are situated around a set of $O(\gamma)$ nooses, where vertices can be repeated at most $O(\gamma)$ times. In such a setting, the number of ways that a partial solution can intersect a middle set is bounded by the number of non-crossing partitions of the boundary-vertices in a fat-connected subset of the surface (see Definition 6.1). By splitting the boundary-vertices that belong to more than one noose, we can assume that these nooses are mutually disjoint. That way, we reduce the problem to the enumeration of the non-crossing partitions of $O(\gamma)$ disjoint nooses containing ℓ vertices, which are $2^{O(\ell)} \cdot \ell^{O(\gamma)} \cdot \gamma^{O(\gamma)}$, as we prove in the following section (Theorem 7.2). Observe that the splitting operation increases the size of the middle sets by at most $O(\gamma)$, therefore $\ell = k + O(\gamma)$ and this yields an upper bound of $2^{O(k)} \cdot k^{O(\gamma)} \cdot \gamma^{O(\gamma)}$ on the size of the tables of the dynamic programming. In Section 7 we use singularity analysis over expressions obtained by the symbolic method to count the number of such non-crossing partitions. Namely, in Sections 7.1 and 7.2 we give a precise estimate of the number of non-crossing partitions in surfaces with boundary. Then we incorporate in Section 7.5 two particularities of surface cut decompositions : Firstly, we deal with the set A of vertices originating from the polyhedral decomposition. These vertices are not situated around the nooses that disconnect the surface into two connected components, and this is why they are treated as *apices* in the enumeration. Secondly, we take into account that, in fact, we need to count the number of non-crossing *packings* rather than the number of non-crossing partitions, as a solution may not intersect *all* the vertices of a middle set, but only a subset. The combinatorial results of Section 7 are of interest by themselves, as they are a natural extension to higher-genus surfaces of the classical non-crossing partitions in the plane, which are enumerated by the Catalan numbers (see e.g. [17]).

7 Non-crossing Partitions in Surfaces with Boundary

In this section we obtain upper bounds for non-crossing partitions in surfaces with boundary. The concept of a non-crossing partition in a general surface is not as simple as in the case of the disk, and must be defined carefully. In fact, the study of non-crossing partitions is a particular case of the study of hypermaps [7]. These objects can be also viewed as bipartite maps. In Section 7.1 we set up our notation. In Section 7.2 we obtain a tree-like structure that provides a way to obtain asymptotic estimates. These estimates are obtained in detail in Sections 7.3 and 7.4, where we employ as a main tool map enumeration techniques. To conclude, in Section 7.5 we extend the enumeration to two more general families.

7.1 2-zone decompositions and non-crossing partitions

Let Σ be a surface with boundary. A *2-zone decomposition* of Σ is a decomposition of Σ where all vertices lay in the boundary of Σ and there is a coloring of the faces using 2 colors (black and white) such that every vertex is incident (possibly more than once) with a unique black face. Black faces are also called *blocks*. A 2-zone decomposition is *regular* if every block is contractible. All 2-zone decompositions are *rooted* : every connected component of the boundary of Σ is edge-rooted. We denote by $\mathcal{S}_\Sigma(k)$, $\mathcal{R}_\Sigma(k)$ the set of general and regular 2-zone decompositions of Σ with k vertices, respectively. A 2-zone decomposition s over Σ defines a non-crossing partition $\pi_\Sigma(s)$ over the set of vertices. Let $\Pi_\Sigma(k)$ be the set of non-crossing partitions of Σ with k vertices. The main objective in this section consists in obtaining bounds for $|\Pi_\Sigma(k)|$. The critical observation is that each non-crossing partition is defined by a 2-zone decomposition. Consequently, $|\Pi_\Sigma(k)| \leq |\mathcal{S}_\Sigma(k)|$. The strategy to enumerate this second set consists in reducing the enumeration to simpler families of 2-zone decompositions. More specifically, Proposition 7.1 shows that it is sufficient to study regular decompositions. First we need a definition and a basic topological lemma.

Let Σ_1 and Σ_2 be surfaces with boundary, possibly not connected. We write $\Sigma_2 \subset \Sigma_1$ if there exists a continuous injection $i : \Sigma_2 \hookrightarrow \Sigma_1$ such that $i(\Sigma_2)$ is homeomorphic to Σ_2 . If s is a 2-zone decomposition of Σ_2 and $\Sigma_2 \subset \Sigma_1$, then the injection i induces a 2-zone decomposition $i(s)$ on Σ_1 such that $\pi_{\Sigma_2}(s) = \pi_{\Sigma_1}(i(s))$. In other words, all 2-zone decompositions over Σ_2 can be realized on a surface Σ_1 which contains Σ_2 . Consequently, informally speaking, $\Pi_{\Sigma_2}(k) \subseteq \Pi_{\Sigma_1}(k)$ if $\Sigma_2 \subset \Sigma_1$.

Lemma 7.1 *Let m^* be a regular 2-zone decomposition of Σ_1 . Let $\Sigma_1 \subset \Sigma$. Then m^* defines a regular 2-zone decomposition m over Σ such that $\pi_{\Sigma_1}(m^*) = \pi_\Sigma(m)$.*

Proof: Let $i : \Sigma_1 \hookrightarrow \Sigma$ be the corresponding injective application, and consider $m = i(m^*)$. In particular, a block π of m^* is topologically equivalent to the block $i(\pi) : i$ is a homeomorphism between Σ and $i(\Sigma)$. Hence $i(\pi)$ is an open contractible set and m is regular. \square

Proposition 7.1 *Let $s \in \mathcal{S}_\Sigma$ be a 2-zone decomposition of Σ and let $\pi_\Sigma(s)$ be the associated non-crossing partition. Then there exists a regular 2-zone decomposition $m \in \mathcal{R}_\Sigma$ such that $\pi_\Sigma(s) = \pi_\Sigma(m)$.*

Proof: The basic idea is the construction of a finite sequence of 2-zone decompositions $s_0 = s, s_1, \dots, s_t = m$, such that $\pi_\Sigma(s_0) = \dots = \pi_\Sigma(s_t)$ and $s_t = m$ is regular. First, consider a non-contractible block f of s . Suppose that the boundary of f consists of more than one connected component. We define the operation of *joining boundaries* as follows : let l be a path that joins a vertex v in one component of the boundary of f with a vertex u in another component. This path exists because f is a face. Consider also a pair of paths l_1, l_2 that joins these two vertices around the initial path l , as illustrated in Figure 2. We define the face f' as the one obtained from f by deleting the face defined by l_1 and l_2 which contains l . Let s_1 be the resulting 2-zone decomposition. Observe that the number of connected components of the boundary of f' is the same as for f minus one, and that $\pi_\Sigma(s) = \pi_\Sigma(s_1)$. We can apply this argument over f as many times as the number of components of the boundary of f . At the end, we obtain a 2-zone decomposition s_{p_1} such that $\pi_\Sigma(s) = \pi_\Sigma(s_1) = \dots = \pi_\Sigma(s_{p_1})$.

Suppose now that the boundary of the block f has one connected component. Additionally, in this block there are vertices on its boundary which are incident with f more than once. Let v be a vertex incident $r > 1$ times with f . In this case we define the operation of *cutting vertices* as follows : consider the intersection of a small neighborhood of v with f , and delete vertex v . This intersection has r connected components. We define a face by pasting v with only one of these components, and disconnecting the others from v (see Figure 2). Then the resulting 2-zone decomposition has the same associated non-crossing partition, and v is incident with the corresponding block exactly once.

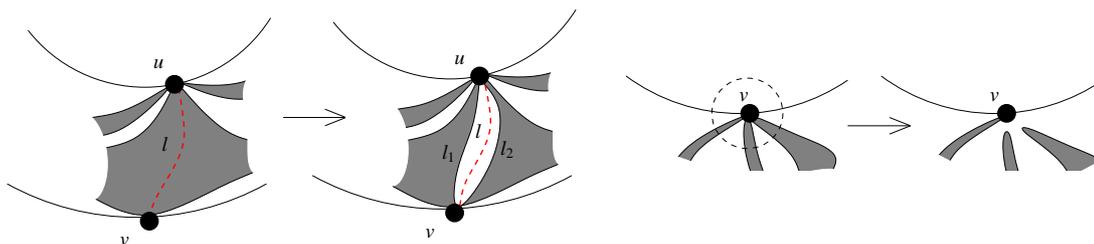


FIG. 2 – The operations of joining boundaries and cutting vertices.

Summarizing, we construct from s a regular 2-zone decomposition in the following way : apply the operation of joining boundaries, and then the operation of cutting vertices. After this, every block has one boundary and each vertex is incident with its corresponding block exactly once. In this case, a block is either contractible or not. If it is not contractible, let \mathbb{S}^1 be a non-contractible cycle, which can be cut using the operator \asymp . For all blocks, the number of times we need to apply this operator is bounded by $\gamma(\Sigma)$. At the end, all blocks are contractible and the resulting surface is $\Sigma_1 \subset \Sigma$. So, the resulting 2-zone decomposition m^* is regular, and then by Lemma 7.1 there exists a regular 2-zone decomposition m over Σ such that $\pi_\Sigma(m) = \pi_{\Sigma_1}(m^*) = \pi_\Sigma(s)$, as claimed. \square

In other words, $|\Pi_\Sigma(k)| \leq |\mathcal{R}_\Sigma(k)|$ for each value of k . Instead of counting $|\mathcal{R}_\Sigma(k)|$, we reduce our study to the family of regular 2-zone decompositions where each face (block or white face) is contractible. The reason is, as we show later, that this subfamily provides the greatest contribution to the asymptotic enumeration. This set is called the set of *irreducible* 2-zone decompositions of Σ ,

and it is denoted by $\mathcal{P}_\Sigma(k)$. Equivalently, an irreducible 2-zone decomposition cannot be realized in a proper surface contained in Σ . The details follow.

A generalization of the notion of irreducibility. We shall provide an equivalent definition and an additional property of irreducible 2-zone decompositions. Recall the definition about inclusion of surfaces stated before Lemma 7.1. We say that a non-crossing partition π_{Σ_1} is *irreducible* in Σ_1 if there is no realization of π_{Σ_1} in a surface Σ_2 such that $\Sigma_2 \subset \Sigma_1$. This definition is compatible with the notion introduced in Section 7.1, as shown in the following lemma :

Lemma 7.2 *Let m be an irreducible 2-zone decomposition of Σ . Then the faces of m are all contractible.*

Proof: We only need to deal with white faces. For a white face whose interior is not an open polygon, there exists a non-contractible cycle \mathbb{S}^1 . Cutting along \mathbb{S}^1 using the operator \succ we obtain a new surface with boundary Σ' such that $\Sigma' \subset \Sigma$ and m is induced in Σ' . As a conclusion, all faces are contractible. □

7.2 Tree-like structures, enumeration, and asymptotic counting

In this section we provide estimates for the number of irreducible 2-zone decompositions, which are obtained directly for the surface Σ . This approach is novel and gives upper bounds close to the exact values. The usual technique consists in reducing the enumeration to surfaces of smaller genus, and returning back to the initial one by topological “pasting” arguments. The main point consists in exploiting tree-like structures of the dual graph associated to an irreducible 2-zone decomposition. The main ideas are inspired by [4], where they are used in the context of map enumeration. For simplicity of the presentation, the construction is explained on the disk. The dual graph of a non-crossing partition on the disk is a tree whose internal vertices are bicolored (black color for blocks). An example of this construction is shown in Figure 3. We use this family of trees (and some related ones) in order to obtain a decomposition of elements of the set $\mathcal{P}_\Sigma(k)$. In Section 7.3 the enumeration of this basic family is done, as well as the enumeration of the related families.

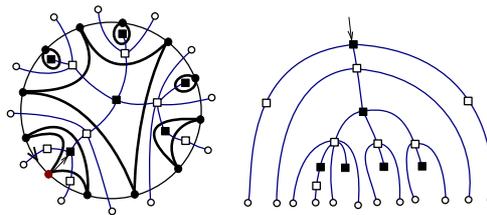


FIG. 3 – A non-crossing partition tree.

The construction for general surfaces is a generalization of the previous one. An example is shown in the leftmost picture of Figure 4. For an element $m \in \mathcal{P}_\Sigma(k)$, denote by M the resulting map

on $\bar{\Sigma}$ (recall the definition of $\bar{\Sigma}$ in Section 2). From M we reconstruct the initial 2-zone decomposition m by pasting vertices of degree 1 which are incident to the same face, and taking the dual map. From M we define a new rooted map on $\bar{\Sigma}$ in the following way : we start deleting recursively vertices of degree 1 which are not roots. Then we continue dissolving vertices of degree 2. The resulting map has $\beta(\Sigma)$ faces and all vertices have degree at least 3 (apart from root vertices, which have degree 1). The resulting map is called the *scheme associated* to M ; we denote it by S_M . See Figure 4 for an example.

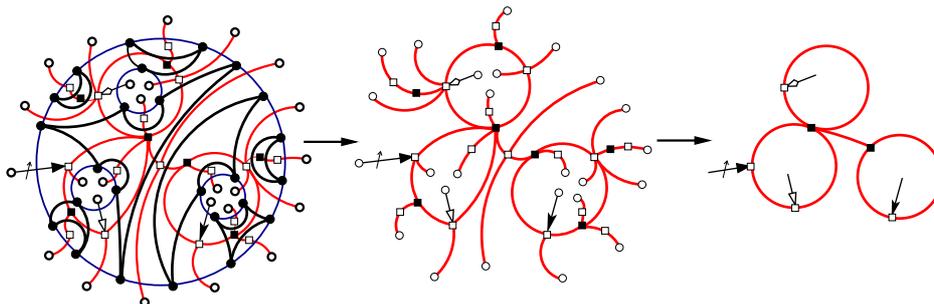


FIG. 4 – The construction of the scheme of an element in \mathcal{P}_Σ . We consider the dual of an irreducible 2-zone decomposition (leftmost figure). After deleting vertices of degree 1 recursively and dissolving vertices of degree 2, we obtain the associated scheme (rightmost figure).

An inverse construction can be done using maps over $\bar{\Sigma}$ and families of plane trees. Using these basic pieces, we can reconstruct all irreducible 2-zone decompositions. The details of this construction can be found in Section 7.3. Exploiting this decomposition and using singularity analysis (see Section 2.4 for the basic definitions), we get the following theorem, whose proof is provided in Section 7.4 (Γ denotes the classical Gamma function [16]) :

Theorem 7.1 *Let Σ be a surface with boundary. Then the number $|\Pi_\Sigma(k)|$ verifies*

$$|\Pi_\Sigma(k)| \leq_{k \rightarrow \infty} \frac{C(\Sigma)}{\Gamma(3/2\gamma(\Sigma) + \beta(\Sigma) - 3)} \cdot k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 4} \cdot 4^k, \quad (2)$$

where $C(\Sigma)$ is a function depending only on Σ that is bounded by $\gamma(\Sigma)^{O(\gamma(\Sigma))}$.

7.3 Enumeration of non-crossing partitions of the disk and related constructions

We first introduce some terminology related to trees that arise as dual graphs of non-crossing partitions on the disk. Then, we use these concepts to obtain the number of non-crossing partitions of the disk with n vertices. At last, we introduce some families of related trees, which are used in the construction of the dual map of a non-crossing partition in a surface of higher genus.

The dual graph of a non-crossing partition is a tree, which is called the (non-crossing partition) tree associated to the non-crossing partition. Vertices of degree 1 (that is, the leaves of the tree) are

called *danglings*. Vertices of the tree are called *block vertices* if they are associated to a block of the non-crossing partition. The remaining vertices are either *non-polygon vertices* or *danglings*. By construction, all vertices adjacent to a polygon vertex are non-polygon vertices. Conversely, each vertex adjacent to a non-polygon vertex is either a block vertex or a dangling. Graphically, we use the symbols \blacksquare for block vertices, \square for non-polygon vertices and \circ for danglings.

Denote by \mathcal{T} the set of non-crossing partitions trees, and let $\mathbf{T} = \mathbf{T}(z, u) = \sum_{k,n>0} \mathbf{t}_{k,m} z^k u^m$ be the corresponding GF. The variable z marks danglings and u marks block vertices. We use also an auxiliary family \mathcal{B} , defined as the set of trees which are rooted at a non-polygon vertex. Let $\mathbf{B} = \mathbf{B}(z, u) = \sum_{k,n>0} \mathbf{b}_{k,m} z^k u^m$ be the associated GF. The next lemma gives the exact enumeration of \mathcal{T} and \mathcal{B} .

Lemma 7.3 *The number of non-crossing trees counted by the number of danglings and block vertices is enumerated by*

$$\mathbf{T}(z, u) = \frac{1 - z(1 - u) - \sqrt{(z(1 - u) - 1)^2 - 4zu}}{2zu}. \quad (3)$$

Furthermore, $\mathbf{B}(z, u) = z\mathbf{T}(z, u)$.

Proof: We establish combinatorial relations between \mathcal{B} and \mathcal{T} , from which we deduce the desired result. First, observe that there is no restriction on the size of the blocks. Hence the degree of every block vertex is arbitrary. This condition is translated symbolically via the following relation : $\mathcal{T} = \blacksquare \times \text{Seq}(\mathcal{B})$. Similarly, \mathcal{B} can be written in the form $\mathcal{B} = \{\circ\} \times \text{Seq}(\mathcal{T} \times \{\circ\})$.

These combinatorial conditions translate using Table 1 into the system of equations

$$\mathbf{T} = \frac{u}{1 - \mathbf{B}}, \quad \mathbf{B} = \frac{z}{1 - z\mathbf{T}}.$$

If we substitute the expression of \mathbf{B} in the first equation, one obtains that \mathbf{T} satisfies the equation $z\mathbf{T}^2 + (z(1 - u) - 1)\mathbf{T} + u = 0$. The valid solution of this equation is (3). Solving the previous system of equations in terms of \mathbf{B} , we obtain that $\mathbf{B} = z\mathbf{T}$, as claimed. \square

Observe that writing $u = 1$, we obtain that $\mathbf{T}(z) = \mathbf{T}(z, 1) = \frac{1 - \sqrt{1 - 4z}}{2z}$, and $\mathbf{B}(z) = \mathbf{B}(z, 1) = z\mathbf{T}(z)$, and we recover the GF for Catalan numbers.

We need also a set of families of trees that are quite related to the previous ones. We call them *double trees*. A double tree is obtained in the following way : consider a path where we concatenate vertices of type \blacksquare with vertices of type \square . A double tree is a tree obtained by pasting on every internal vertex of type \blacksquare a pair of elements of \mathcal{T} (one at each side of the path), and similarly for internal vertices of type \square . We say that a double tree is of type either $\blacksquare - \blacksquare - \square$, or $\square - \square$ depending on the type of the ends of the path. An example for a double tree of type $\blacksquare - \blacksquare$ is shown in Figure 5.

We denote these families by $\mathcal{T}_{\blacksquare-\blacksquare}$, $\mathcal{T}_{\square-\blacksquare}$, and $\mathcal{T}_{\square-\square}$, and the corresponding GF by $\mathbf{T}_1(z, u) = \mathbf{T}_1$, $\mathbf{T}_2(z, u) = \mathbf{T}_2$ and $\mathbf{T}_3(z, u) = \mathbf{T}_3$, respectively. Recall that in all cases z marks danglings and u marks block vertices. A direct application of the symbolic method provides a way to obtain explicit expressions for the previous GFs. The decomposition and the GFs of the three families is summarized in Table 2.

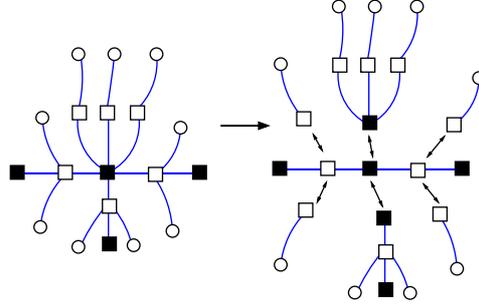


FIG. 5 – A double tree and its decomposition.

Family	Development	Compact expression
$\mathcal{T}_{\square-\blacksquare}$	$1 + \frac{1}{u}\mathbf{B}^2\mathbf{T}^2 + \frac{1}{u^2}\mathbf{B}^4\mathbf{T}^4 + \dots$	$1/(1 - \mathbf{T}^2\mathbf{B}^2/u)$
$\mathcal{T}_{\blacksquare-\blacksquare}$	$\mathbf{B}^2 + \frac{1}{u}\mathbf{B}^4\mathbf{T}^2 + \frac{1}{u^2}\mathbf{B}^6\mathbf{T}^4 + \dots$	$\mathbf{B}^2/(1 - \mathbf{T}^2\mathbf{B}^2/u)$
$\mathcal{T}_{\square-\square}$	$\frac{1}{u}\mathbf{T}^2 + \frac{1}{u^2}\mathbf{B}^2\mathbf{T}^4 + \frac{1}{u^3}\mathbf{B}^4\mathbf{T}^6 + \dots$	$\frac{1}{u}\mathbf{T}^2/(1 - \mathbf{T}^2\mathbf{B}^2/u)$

TAB. 2 – GFs for double trees.

To conclude, the family of *pointed* non-crossing trees \mathcal{T}^\bullet is built pointing a dangling over each tree. In this case, the associated GF is $\mathbf{T}^\bullet = z \frac{\partial}{\partial z} \mathbf{T}$. Similar definitions can be done for the family \mathcal{B} . Pointing a dangling define a unique path between this distinguished dangling and the root of the tree.

7.4 Proof of Theorem 7.1

The steps towards the proof of Theorem 7.1 are developed in Sections 7.4.1, 7.4.2, and 7.4.3. Basically, we start characterizing the combinatorial decomposition in terms of plane trees. This combinatorial decomposition is exploited in Proposition 7.2 of Section 7.4.1 in order to count irreducible 2-zone decompositions. The constant $C(\Sigma)$ is related to the enumeration of cubic maps [3, 21]. Bounds for $C(\Sigma)$ are given in Section 7.4.2 (see Proposition 7.4). Finally, we prove in Section 7.4.3 that the asymptotic of $|\mathcal{R}_\Sigma(k)|$ coincides with the one obtained for irreducible 2-zone decompositions. The argument uses a double induction on the number of boundaries and the genus of the surface, and Lemma 5.3 of Section 5.

7.4.1 Combinatorial decomposition and enumeration

We use the notation and definitions introduced in Section 7.3 (i.e., families of trees, double trees and pointed trees, and the corresponding GFs). To simplify the notation, we denote by $\mathcal{P}_\Sigma(k, m)$ the set of irreducible 2-zone decompositions of Σ with k vertices and m blocks. We write $p_{k,m}^\Sigma$ for

the cardinal of this set. Let $p_k^\Sigma = \sum_{m>0} p_{k,m}^\Sigma$. The GF associated to the numbers $p_{k,m}^\Sigma$ is denoted by $\mathbf{P}_\Sigma(z, u)$. We denote by \mathfrak{S}_Σ the set of rooted maps on $\bar{\Sigma}$ with $\beta(\Sigma)$ faces, whose vertices are bicolored (either \blacksquare or \square) and have degree at least 3. In particular, endpoints of a given edge can have the same color. This notation is used in Sections 7.4.3, 7.5.1, and 7.5.2. Observe that in our framework, each map has $\beta(\Sigma)$ roots, in contrast to the classical theory of enumeration of rooted maps (where a unique root is considered).

Applying Euler's formula for maps on $\bar{\Sigma}$ implies that $|\mathfrak{S}_\Sigma|$ is finite, because the number of faces is fixed. It is also obvious that if S_M is the scheme associated to a map M , then $S_M \in \mathfrak{S}_\Sigma$. These observations provide a way to establish a combinatorial bijection, that can be exploited to obtain the enumeration of \mathcal{P}_Σ . More concretely, each element M can be constructed from an element S of \mathfrak{S}_Σ in the following way :

1. For an edge of S with both end-vertices of type \blacksquare , we paste a double tree of type $\blacksquare - \blacksquare$ along it. Similar operations can be realized for edges with end-vertices $\{\square, \blacksquare\}$ and $\{\square, \square\}$.
2. For a block vertex v of S , not incident with any root, we paste $d(v)$ elements of \mathcal{T} (identifying the roots of trees in \mathcal{T} with v), one in each region determined by consecutive half-edges.
3. For a set of roots with an end-point in the same block vertex v , we paste an element of \mathcal{T}^\bullet along each one of the roots (the marked leaf determines which is the dangling root). Over v we paste trees of \mathcal{T} as we have done in the previous case. We do not paste trees between a root and a half-edge of S . A similar operation is done if the vertex is of type \square .

This construction is shown for a concrete example in Figure 6. Let us introduce some notation.

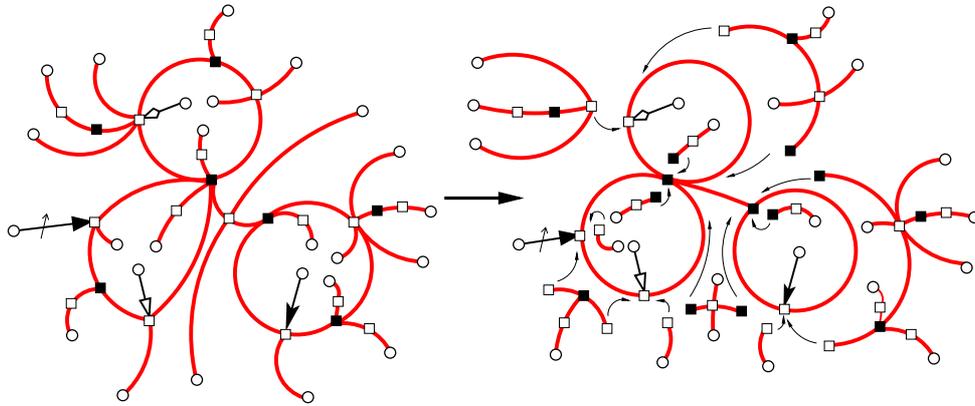


FIG. 6 – The decomposition into bicolored trees and the associated scheme.

Consider an element S of \mathfrak{S}_Σ . Let $v_1(S), v_2(S)$ be the set of block vertices and non-polygon vertices of S , respectively. Write $B(S), W(S)$ for the number of roots which are incident with a vertex of type \blacksquare and \square , respectively. In particular, $B(S) + W(S) = \beta(\Sigma)$. Denote by $e_1(S)$ the number of edges in S of type $\blacksquare - \blacksquare$. We similarly define $e_2(S)$ and $e_3(S)$ for edges of type $\square - \blacksquare$ and $\square - \square$, respectively.

Observe that $e_1(S) + e_2(S) + e_3(S) + B(S) + W(S)$ is the number of edges of S , that is $e(S) = |E(S)|$. For a vertex x of S , denote by $r(x)$ the number of roots which are incident with it.

The previous decomposition provides a direct way to obtain the desired enumeration.

Proposition 7.2 *Let Σ be a surface with boundary. Then the coefficient $[z^k]\mathbf{P}_\Sigma(z, 1)$ has an asymptotic expansion of the form*

$$[z^k]\mathbf{P}_\Sigma(z, 1) = p_k^{\Sigma} =_{k \rightarrow \infty} \frac{C(\Sigma)}{\Gamma(-3\chi(\Sigma)/2 + \beta(\Sigma))} k^{-3\chi(\Sigma)/2 + \beta(\Sigma) - 1} 4^k (1 + \mathcal{O}(k^{-1/2})), \quad (4)$$

where $C(\Sigma)$ is a function depending only on Σ .

Proof: According to the previous observations, $\mathbf{P}_\Sigma(z, u)$ can be written in the following form : for each $S \in \mathfrak{S}_\Sigma$, we replace edges (not roots) with double trees, roots with pointed trees, and vertices with sets of trees. More concretely, the GF we obtain is

$$\sum_{S \in \mathfrak{S}_\Sigma} u^{|v_1(S)|} \mathbf{T}_1^{e_1(S)} \mathbf{T}_2^{e_2(S)} \mathbf{T}_3^{e_3(S)} \left(\frac{\mathbf{T}}{u} \right)^{\sum_{x \in v_1(S)} (d(x) - 2r(x))} \mathbf{B}^{\sum_{y \in v_2(S)} (d(y) - 2r(y))} \left(\frac{\mathbf{T}^\bullet}{u} \right)^{B(S)} (\mathbf{B}^\bullet)^{W(S)}. \quad (5)$$

Observe that terms \mathbf{T} and \mathbf{T}^\bullet appear divided by u . The reason is that we paste non-crossing trees through the root, which is a block vertex. In order to do it, we delete the corresponding block vertex, we paste the trees identifying their roots without counting the root, and finally we add the total number of block vertices (thus the term $u^{|v_1(S)|}$). To obtain the asymptotic behavior in terms of the number of danglings, we write $u = 1$ in Equation (5). To study the resulting GF, we need the expression of each factor of Equation (5) when we write $u = 1$. In Table 3 all the expressions are shown. This table is built from the expressions for \mathbf{T} and \mathbf{B} deduced in Lemma 7.3 and the expressions for double trees in Table 2. The GF in Equation (5) is a finite sum (a total of $|\mathfrak{S}_\Sigma|$

GF	Expression
$\mathbf{T}_1(z)$	$1/16(1 - 4z)^{-1/2} - 1/8(1 - 4z)^{1/2} + 1/16(1 - 4z)^{3/2}$
$\mathbf{T}_2(z)$	$1/4(1 - 4z)^{-1/2} + 1/2 + (1 - 4z)^{1/2}$
$\mathbf{T}_3(z)$	$z^2 \left(1/16(1 - 4z)^{-1/2} - 1/8(1 - 4z)^{1/2} + 1/16(1 - 4z)^{3/2} \right)$
$\mathbf{T}(z)$	$1/(2z)(1 - (1 - 4z)^{1/2})$
$\mathbf{B}(z)$	$1/2 \left(1 - (1 - 4z)^{1/2} \right)$
$\mathbf{T}^\bullet(z)$	$1/z(1 - 4z)^{-1/2} - 1/(2z^2)(1 - (1 - 4z)^{-1/2})$
$\mathbf{B}^\bullet(z)$	$(1 - 4z)^{-1/2}$

TAB. 3 – Univariate GF for all families of trees.

terms), so its singularity is located at $z = 1/4$ (since each addend has a singularity at this point). For each choice of S ,

$$\mathbf{T}(z, 1)^{\sum_{x \in v_1(S)} (d(x) - 2r(x))} \mathbf{B}(z, 1)^{\sum_{y \in v_2(S)} (d(y) - 2r(y))} = \sum_{n=0}^{f(S)} f_n(z) (1 - 4z)^{n/2}, \quad (6)$$

where the positive integer $f(S)$ depends only on S , $f_n(z)$ are functions analytic at $z = 1/4$, and $f_0(z) \neq 0$ at $z = 1/4$. For the other multiplicative terms, we obtain

$$\mathbf{T}_1(z, 1)^{e_1(S)} \mathbf{T}_2(z, 1)^{e_2(S)} \mathbf{T}_3(z, 1)^{e_3(S)} \mathbf{T}^\bullet(z, 1)^{B(S)} \mathbf{B}^\bullet(z, 1)^{W(S)} = G_S(z)(1 - 4z)^{-\frac{e(S)}{2}} + \dots, \quad (7)$$

where $G_S(z)$ is an analytic function at $z = 1/4$. The reason of this fact is that each GF in the previous formula can be written in the form $p(z)(1 - 4z)^{-1/2} + \dots$, where $p(z)$ is a function analytic at $z = 1/4$, and $e_1(S) + e_2(S) + e_3(S) + B(S) + W(S)$ is the total number of edges. Multiplying Equation (6) and Expression (7) we recover the contribution of a map S in $\mathbf{P}_\Sigma(z, 1)$. More concretely, the contribution of a single map S to Equation (5) can be written in the form

$$g_S(z)(1 - 4z)^{-e(S)/2} + \dots,$$

where $g_S(z)$ is an analytic function at $z = 1/4$. From Equation (1), the maps giving the greatest contribution to the asymptotic of p_k^Σ are the ones which maximize the value of $e(S)$. Applying Euler's formula (recall that all maps in \mathfrak{S}_Σ have $\beta(\Sigma)$ faces) on $\bar{\Sigma}$ gives that these maps are the ones where each vertex have degree 3 (i.e., cubic maps). In particular, cubic maps with $\beta(\Sigma)$ faces and $\beta(\Sigma)$ roots have $2\beta(\Sigma) - 3\chi(\Sigma)$ edges. Hence, as a consequence of the Transfer Theorem for singularity analysis, the singular expansion of $\mathbf{P}_\Sigma(z, 1)$ at $z = 1/4$ is

$$\mathbf{P}_\Sigma(z, 1) =_{z \rightarrow 1/4} C(\Sigma)(1 - 4z)^{3\chi(\Sigma)/2 - \beta(\Sigma)} \left(1 + \mathcal{O}((1 - 4z)^{1/2})\right), \quad (8)$$

where $C(\Sigma) = \sum_{S \in \mathfrak{S}_\Sigma} g_S(1/4)$. Applying the Transfer Theorem in this expression yields the claimed result. \square

7.4.2 Bounding $C(\Sigma)$ in terms of cubic maps

In this section we obtain bounds for $C(\Sigma)$. A more refined analysis over functions $g_S(z)$ provides upper bounds for $C(\Sigma)$. This is done in the following proposition :

Proposition 7.3 *The function $C(\Sigma)$ defined in Proposition 7.2 satisfies*

$$C(\Sigma) \leq 2^{\beta(\Sigma)} |\mathfrak{S}_\Sigma|. \quad (9)$$

Proof: For each $S \in \mathfrak{S}_\Sigma$, we obtain bounds for $g_S(1/4)$. We use Table 4, which is a simplification of Table 3. We are only concerned now about the constant term of each GF. Table 4 brings the following information : the greatest contribution from double trees, trees, and families of pointed trees comes from $\mathcal{T}_{\square-\blacksquare}$, \mathcal{T} , and \mathcal{T}^\bullet , respectively. The constants are $1/4$, 2 , and 4 , respectively. Each cubic map has $2\beta(\Sigma) - 3\chi(\Sigma)$ edges ($\beta(\Sigma)$ of them being roots) and $\beta(\Sigma) - 2\chi(\Sigma)$ vertices ($\beta(\Sigma)$ of them being incident with roots). This characterization provides the following upper bound for $g_S(1/4)$:

$$g_S(1/4) \leq \left(\frac{1}{4}\right)^{2\beta(\Sigma) - 3\chi(\Sigma) - \beta(\Sigma)} 2^{-3 \cdot 2\chi(\Sigma) + \beta(\Sigma)} 4^{\beta(\Sigma)} = 2^{\beta(\Sigma)}. \quad (10)$$

\square

GF	Expression	Developement at $z = 1/4$
$\mathbf{T}_1(z)$	$1/16(1-4z)^{-1/2} + \dots$	$1/16(1-4z)^{-1/2} + \dots$
$\mathbf{T}_2(z)$	$1/4(1-4z)^{-1/2} + \dots$	$1/4(1-4z)^{-1/2} + \dots$
$\mathbf{T}_3(z)$	$z^2/16(1-4z)^{-1/2} + \dots$	$1/256(1-4z)^{-1/2} + \dots$
$\mathbf{T}(z)$	$1/(2z) + \dots$	$2 + \dots$
$\mathbf{B}(z)$	$1/2 + \dots$	$1/2 + \dots$
$\mathbf{T}^\bullet(z)$	$1/z(1-4z)^{-1/2} + \dots$	$4(1-4z)^{-1/2} + \dots$
$\mathbf{B}^\bullet(z)$	$(1-4z)^{-1/2}$	$(1-4z)^{-1/2}$

TAB. 4 – A simplification of Table 3 used in Proposition 7.2.

The value of \mathfrak{S}_Σ can be bounded using the results in [3, 21]. Indeed, Gao shows in [21] that the number of rooted cubic maps with n vertices in an orientable surface of genus⁵ g is asymptotically equal to

$$t_g \cdot n^{5(g-1)/2} \cdot (12\sqrt{3})^n,$$

where the constant t_g tends to 0 as g tends to ∞ [3]. A similar result is also stated in [21] for non-orientable surfaces. By duality, the number of rooted cubic maps in a surface $\bar{\Sigma}$ of genus $\chi(\Sigma)$ with $\beta(\Sigma)$ faces is asymptotically equal to $t_{\chi(\Sigma)} \cdot \beta(\Sigma)^{5(\chi(\Sigma)-1)/2} \cdot (12\sqrt{3})^{\beta(\Sigma)}$. This value is clearly bounded by $\gamma(\Sigma)^{O(\gamma(\Sigma))}$.

To conclude, we observe that the elements of \mathfrak{S}_Σ are obtained from rooted cubic maps with $\beta(\Sigma)$ faces by adding a root on each face different from the root face. Observe that each edge is incident with at most two faces, and that the total number of edges is $-3\chi(\Sigma)$. Consequently, the number of ways of rooting a cubic map with $\beta(\Sigma) - 1$ unrooted faces is bounded by $\binom{-6\chi(\Sigma)}{\beta(\Sigma)-1}$, which is bounded by $\gamma(\Sigma)^{O(\gamma(\Sigma))}$.

By the above discussion, the following proposition holds.

Proposition 7.4 *The constant $C(\Sigma)$ verifies*

$$C(\Sigma) \leq t_{\chi(\Sigma)} \cdot \beta(\Sigma)^{5(\chi(\Sigma)-1)/2} \cdot (12\sqrt{3})^{\beta(\Sigma)} \binom{-6\chi(\Sigma)}{\beta(\Sigma)-1} 2^{\beta(\Sigma)}.$$

In particular, $C(\Sigma) = \gamma(\Sigma)^{O(\gamma(\Sigma))}$.

Combining Propositions 7.2 and 7.4, we obtain that

$$P_k^\Sigma \leq_{k \rightarrow \infty} \frac{C(\Sigma)}{\Gamma(3/2\gamma(\Sigma) + \beta(\Sigma) - 3)} \cdot k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 4} \cdot 4^k, \quad (11)$$

where $C(\Sigma) = \gamma(\Sigma)^{O(\gamma(\Sigma))}$ is a function depending only on Σ .

⁵the genus $g(\Sigma)$ of an orientable surface Σ is defined as $g(\Sigma) = \gamma(\Sigma)/2$ (see [24]).

7.4.3 Irreducibility vs reducibility

For conciseness, we use the notation $a(\Sigma)$ to denote the constant term which appears in all the asymptotic expressions in Section 7.3.

For a non-irreducible regular element s of \mathcal{R}_Σ (recall Lemma 7.2) there is a non-contractible cycle \mathbb{S}^1 contained in a white 2-dimensional region of s . Additionally, s induces a regular 2-zone decomposition over the surface $\Sigma \succ \mathbb{S}^1 = \Sigma'$, which can be irreducible or not. By Lemma 7.1, each element of $\mathcal{R}_{\Sigma'}$ defines an element on \mathcal{R}_Σ . To prove that irreducible 2-zone decompositions over Σ give the maximal contribution to the asymptotic, we apply a double induction argument on the pair $(\gamma(\Sigma), \beta(\Sigma))$. The critical point is the initial step, which corresponds to $\gamma(\Sigma) = 0$:

Proposition 7.5 *Let Σ be a surface obtained from the sphere deleting β disjoint disks. Then*

$$|\mathcal{R}_\Sigma(k)| \leq_{k \rightarrow \infty} |\mathcal{P}_\Sigma(k)|.$$

Proof: Induction on β . The case $\beta = 1$ corresponds to the disk. We deduced in Section 7.3 the exact expression for $\mathbf{P}_\Sigma(z, u)$ (see Equation (3)). In this case the equality $|\mathcal{R}_\Sigma(k)| = |\mathcal{P}_\Sigma(k)|$ holds for every value of k . Let us consider now the case $\beta = 2$, which corresponds to the cylinder. From Equation (11), the number of irreducible 2-zone decompositions over the cylinder verifies

$$|\mathcal{P}_\Sigma(k)| \underset{k \rightarrow \infty}{=} a(\Sigma) \cdot k \cdot 4^k (1 + O(k^{-1/2})). \quad (12)$$

Let us calculate upper bounds for the asymptotic of non-irreducible 2-zone decompositions on a cylinder. A non-contractible cycle \mathbb{S}^1 on a cylinder separates it into a pair of cylinders. In other words $\Sigma' = \Sigma \succ \mathbb{S}^1$ is a pair of disks. The asymptotic in this case is of the form $[z^k] \mathbf{T}(z, 1)^2 \underset{k \rightarrow \infty}{=} O(k^{-3/2} 4^k)$. The subexponential term in Equation (12) is greater, so the claim of the proposition holds for $\beta = 1$.

Let us proceed to apply the inductive step. Let $\beta > 1$ be the number of boundaries of Σ . A non-contractible cycle \mathbb{S}^1 always separates Σ into two connected components, namely Σ_1 and Σ_2 . Let $\beta_1, \beta_2 < \beta$ be the number of boundaries of Σ_1 and Σ_2 , respectively. By induction hypothesis,

$$|\mathcal{R}_{\Sigma_j}(k)| \leq_{k \rightarrow \infty} |\mathcal{P}_{\Sigma_j}(k)|,$$

for $j = 1, 2$. Consequently, we only need to deal with irreducible decompositions of Σ_1 and Σ_2 . The GF of 2-zone regular decompositions that reduces to decompositions over Σ_1 and Σ_2 has the same asymptotic as $\mathbf{P}_{\Sigma_1}(z, 1) \cdot \mathbf{P}_{\Sigma_2}(z, 1)$. The estimate of its coefficients is

$$[z^k] \mathbf{P}_{\Sigma_1}(z, 1) \cdot \mathbf{P}_{\Sigma_2}(z, 1) \leq a(\Sigma_1) \cdot a(\Sigma_2) [z^k] (1 - 4z)^{-5/2\beta_1+3} \cdot (1 - 4z)^{-5/2\beta_2+3} \underset{k \rightarrow \infty}{=} O(k^{5/2\beta-7} \cdot 4^k).$$

Consequently, the above term is smaller than p_k^Σ when k is large enough (the value is of the form $(k^{5/2\beta-4} 4^k)$), and does not depend on how Σ is cut. \square

The next step is to adapt the previous argument to surfaces of genus greater than 0. Let Σ be a surface with boundary and Euler genus $\gamma(\Sigma)$. Consider a non-contractible cycle \mathbb{S}^1 and the resulting surface $\Upsilon = \Sigma \succ \mathbb{S}^1$. Two situations can occur :

1. Υ is connected and $\beta(\Upsilon) = \beta(\Sigma)$. In this case, the Euler genus has been decreased by either 1 if the cycle is one-sided or by 2 if the cycle is two-sided. This result appears as Lemma 4.2.4 in [24].
2. The resulting surface is not connected, $\Upsilon = \Upsilon_1 \sqcup \Upsilon_2$. In this case, the total number of boundaries is $\beta(\Upsilon) = \beta(\Upsilon_1) + \beta(\Upsilon_2)$. By Lemma 5.3, $\gamma(\Sigma) = \gamma(\Upsilon_1) + \gamma(\Upsilon_2) - 2$.

The induction argument distinguishes between these two cases : if $\Upsilon = \Sigma \times \mathbb{S}^1$ is connected, by induction on the genus, $|\mathcal{R}_\Upsilon(k)| \leq_{k \rightarrow \infty} |\mathcal{P}_\Upsilon(k)|$. Additionally, by Expression (11), an upper bound for $|\mathcal{P}_\Upsilon(k)|$ is

$$[z^k] \mathbf{P}_\Upsilon(z, 1) = a(\Upsilon) \cdot k^{3/2\gamma(\Upsilon) + \beta(\Upsilon) - 4} \cdot 4^k (1 + \mathcal{O}(k^{-1/2})) \underset{k \rightarrow \infty}{=} \mathcal{O}\left(k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 4} \cdot 4^k\right).$$

If Υ is not connected, then $\Upsilon = \Upsilon_1 \sqcup \Upsilon_2$, $\beta(\Sigma) = \beta(\Upsilon) - 2 = \beta(\Upsilon_1) + \beta(\Upsilon_2)$, and $\gamma(\Sigma) = \gamma(\Upsilon_1) + \gamma(\Upsilon_2)$. Again, by induction hypothesis we only need to look at the irreducible ones. Consequently,

$$[z^k] \mathbf{P}_{\Upsilon_1}(z, 1) \mathbf{P}_{\Upsilon_2}(z, 1) = a(\Upsilon_1) \cdot a(\Upsilon_2) [z^k] (1 - 4z)^{-3/2(\gamma(\Upsilon_1) + \gamma(\Upsilon_2)) - (\beta(\Upsilon_1) + \beta(\Upsilon_2)) + 6}.$$

The exponent of $(1 - 4z)$ can be written as $-3/2\gamma(\Sigma) - \beta(\Sigma) + 6$. Consequently, the value $[z^k] \mathbf{P}_{\Upsilon_1}(z, 1) \mathbf{P}_{\Upsilon_2}(z, 1)$ is bounded, for k large enough, by

$$k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 6 - 1} \cdot 4^k = k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 7} \cdot 4^k = \mathcal{O}\left(k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 4} \cdot 4^k\right).$$

Hence the contribution is smaller than the one given by $|\mathcal{P}_\Sigma(k)|$, as claimed.

7.5 Additional constructions

So far, we enumerated families of non-crossing partitions with boundary. Here, we first deal in Section 7.5.1 with a set of additional vertices that play the role of *apices* (cf. the last paragraph of Section 6). Secondly, we show in Section 7.5.2 how to extend the enumeration from non-crossing partitions to non-crossing packings. In both cases, we show that the modifications over generating functions (GFs for short) do not depend on the surface Σ where non-crossing partitions are considered. The analysis consists in symbolic manipulation of GFs and application of singularity analysis over the resulting expressions. Finally, we provide the resulting asymptotic estimation in Section 7.5.3.

The first problem can be stated in general as follows : for a sequence of positive numbers $\{p_{k,r}\}$, such that we know the GF $\sum_{k,r>0} p_{k,r} z^k u^r$, we want to estimate the value of $\sum_{r=1}^k r^l p_{k,r}$ for a fixed value l . This problem arises from the fact that we have a set of vertices (the *apices*), such that every vertex of the set can be associated to an arbitrary block of a non-crossing partition. The details of the analysis of this problem are done in Section 7.5.1. Basically, this problem only introduces a variation in the subexponential term of the asymptotic stated in Theorem 7.1.

The second problem consists in generalizing from non-crossing partitions to non-crossing packings. In other words, for a fixed number of k vertices, fix an arbitrary subset of $i \leq k$ vertices, and consider the set of non-crossing partitions on Σ on this set of i vertices. This value is precisely $|\Pi_\Sigma(i)|$. Among the total set of k vertices, this set of i vertices can be chosen in $\binom{k}{i}$ ways. Hence, we want to

estimate the sum $\sum_{i=0}^k \binom{k}{i} |\Pi_{\Sigma}(i)|$. Observe that this construction is quite close to Bell numbers, which count the number of ways a set of k elements can be partitioned into nonempty subsets. The details of the analysis can be found in Section 7.5.2. In this case, a combinatorial trick (Lemma 7.4) shows that the modification only affects the base of the exponential term.

7.5.1 Dealing with a set of “apices”

Due to the definition of surface cut decomposition, we need to modify the family \mathcal{P}_{Σ} of irreducible 2-zone decompositions in the following way : consider a set of l vertices $\{\bar{1}, \bar{2}, \dots, \bar{l}\} = [\bar{l}]$ disjoint from the set of vertices over Σ . This set of vertices is called set of *apices*. For every value of k we want to count the number of pairs (s_r, f) , where $s_r \in \mathcal{P}_{\Sigma}(k)$ has r blocks, and f is an arbitrary application $f : [\bar{l}] \rightarrow [r]$. The number of such pairs is $\sum_{r=1}^k r^l p_{k,r}^{\Sigma}$ (recall that the number of irreducible 2-zone decompositions with k vertices and r blocks is $p_{k,r}^{\Sigma}$, and the associated GF is $\mathbf{P}_{\Sigma}(z, u)$). The aim of this section is to obtain estimates for this sum. This problem can be stated in the following equivalent way : let $\mathbf{F}(z, u)$ be a GF with expansion $\mathbf{F}(z, u) = \sum_{k,r \geq 0} f_{k,r} z^k u^r$, such that $[z^k u^r] \mathbf{F}(z, u) = 0$ if $r > k$. For a non-negative integer l , we want to estimate the sum

$$\sum_{r=0}^k r^l f_{k,r} = [z^k] \sum_{k \geq 0} \sum_{r=0}^{\infty} r^l f_{k,r} z^k u^r \Big|_{u=1},$$

for k large enough. Let Θ be the pointing operator on the second variable : $\Theta \mathbf{F}(z, u) = u \frac{\partial}{\partial u} \mathbf{F}(z, u) = u \mathbf{F}_u(z, u)$. Applying l times the operator Θ over $\mathbf{F}(z, u)$ gives $\Theta^l \mathbf{F}(z, u) = \sum_{k,m \geq 0} m^l f_{k,r} z^k u^m$, so our problem consists in estimating $[z^k] \Theta^l \mathbf{F}(z, u) \Big|_{u=1}$. The strategy we use to obtain the estimate consists in simplifying this expression up to a function from which we know to get the asymptotic. Firstly, observe that Θ^l can be written as $\sum_{i=1}^l q_i(u) \frac{\partial^i}{\partial u^i}$, where $q_i(u)$ is a polynomial on u . For $i = l$, the value of $q_l(u)$ is u^l . We show that the greatest contribution to the enumeration comes from the term $i = l$. As a consequence, we only need to deal with $u^l \frac{\partial^l}{\partial u^l} \mathbf{F}(z, u)$. To do this, it is also convenient to observe the following : for $b < a$ positive real numbers, the asymptotic of $(1 - 4z)^{-a}$ is greater than the one for $(1 - 4z)^{-b}$: from the Transfer Theorem for singularity analysis (Equation (1)), both GFs have an exponential growth of the form 4^k . However, their asymptotic growth is not the same : while the first one has a subexponential growth of the form k^{a-1} , the second one is of the form k^{b-1} , which is smaller. Generalizing this to a linear combination of terms of the form $(1 - 4z)^{-a_i}$, where a_i is a positive real number, the asymptotic of the whole function comes from the value a_i with the greatest modulus.

Let us return to study $\mathbf{P}_{\Sigma}(u, z)$. Observe that GFs for double trees can be factorized in the following way (consult Table 2) :

$$\frac{\mathbf{G}(z, u)}{1 - \frac{1}{u} \mathbf{T}^2 \mathbf{B}^2 / u} = \frac{-2uz^2 \mathbf{G}(z, u)}{((u+1)z-1) \sqrt{(z(1-u)-1)^2 - 4zu} + z^2(u-1)^2 - 2z(u+1) + 1},$$

where $\mathbf{G}(z, u)$ is either 1, \mathbf{T}^2/u or \mathbf{B}^2 . For conciseness, write $\mathbf{f} = (z(1-u)-1)^2 - 4zu$, $\mathbf{g} = ((u+1)z-1)$ and $\mathbf{h} = z^2(u-1)^2 - 2z(u+1) + 1$. The previous formula can be written in the

form $-2uz^2\mathbf{G}(z, u)/(\mathbf{h} + \mathbf{g}\sqrt{\mathbf{f}})$. Additionally, $\mathbf{g}(z, 1) = 2z - 1$, $\mathbf{f}(z, 1) = 1 - 4z$ and $\mathbf{h}(z, 1) = 1 - 4z$. For $u = 1$, the smallest singularity of the function is located at $z = 1/4$, where function $\sqrt{\mathbf{f}}$ ceases to be analytic. Consequently, the source of the singularity on a double tree comes exclusively from the term $\sqrt{\mathbf{f}}$. Furthermore, when we write $u = 1$, the smallest singularity of every derivative (with respect to u) of $-2uz^2\mathbf{G}(z, u)/(\mathbf{h} + \mathbf{g}\sqrt{\mathbf{f}})$ is located at $z = 1/4$, because the denominator is always the same (possibly with a greater exponent). A similar argument applies to the families of pointed trees (same behavior, $\mathbf{f}^{-1/2}$).

Taking into account this, and rationalizing the previous expressions, Expression (5) can be written in the form

$$\mathbf{P}_\Sigma(z, u) = \sum_{S \in \mathfrak{S}_\Sigma} (\mathbf{g}_S(z, u)\mathbf{f}^{-e(S)/2} + \dots),$$

where “...” means that the exponent of the other terms is smaller in modulus (and they give smaller contributions to the asymptotic enumeration). Observe that $\mathbf{g}(z, u)$ is analytic at $(z, u) = (1/4, 1)$, and satisfies that $\mathbf{g}_S(z, 1) = g_S(z)$. This presentation for $\mathbf{P}_\Sigma(z, u)$ is the correct one to deal with the operator Θ^l : observe that the greatest contribution (using the Transfer Theorem) comes from cubic maps, which are the ones with maximize the number of edges (i.e., the value $e(S) = 3\gamma(\Sigma) + 2\beta(\Sigma) - 6$). For conciseness on the formulas, until the end of this section we write $e = 3\gamma(\Sigma) + 2\beta(\Sigma) - 6$.

We need to study the derivative $\frac{\partial^l}{\partial u^l} (\mathbf{g}_S(z, u)\mathbf{f}^{-e/2})$, which is the main contribution of each cubic map. When we apply this derivative over $\mathbf{g}_S(z, u)\mathbf{f}^{-e/2}$, the greatest contribution comes from $\mathbf{g}_S(z, u)u^l \frac{\partial^l}{\partial u^l} \mathbf{f}^{-e/2}$, because this term maximizes the exponent (in modulus) of the singular term. In this case, the singular term with greatest exponent corresponds to

$$u^l \mathbf{g}_S(z, u) \frac{(-1)^l \Gamma(e/2 + l)}{\Gamma(e/2)} \frac{(\mathbf{f}_u(z, u))^l}{\mathbf{f}(z, u)^{e/2+l}},$$

where $\mathbf{f}_u(z, u)$ is the derivative of \mathbf{f} with respect to u . Writing $u = 1$, the previous expression is simplified into

$$g_S(z) \frac{\Gamma(e/2 + l)}{\Gamma(e/2)} \frac{(2z)^l}{(1 - 4z)^{e/2+l}}.$$

To estimate the value of the k -th coefficient of the previous GF, we apply the Transfer Theorem for singularity analysis (Equation (1)), obtaining

$$g_S(1/4) \frac{1}{\Gamma(e/2)} 2^{-l} \cdot k^{e/2+l-1} \cdot 4^k (1 + O(k^{-1/2})).$$

To conclude, recall that this above term is the contribution of a single cubic map. Summing over all cubic maps, we obtain the following proposition:

Proposition 7.6 *Let $p_{k,r}^\Sigma$ be the number of irreducible 2-zone decompositions of Σ . For a fixed positive integer l , the following asymptotic approximation holds:*

$$\sum_{r=1}^k r^l p_{k,r}^\Sigma \underset{k \rightarrow \infty}{=} \frac{C(\Sigma) \cdot 2^{-l}}{\Gamma(3\gamma(\Sigma)/2 + \beta(\Sigma) - 3)} \cdot k^{3\gamma(\Sigma)/2 + \beta(\Sigma) - 4 + l} \cdot 4^k (1 + O(k^{-1/2})), \quad (13)$$

where an upper bound for $C(\Sigma)$ is stated in Proposition 7.3.

7.5.2 Bell structures : from partitions to packings

For a fixed number of k vertices, fix an arbitrary subset of $i \leq k$ vertices, and consider the set of non-crossing partitions over Σ using this set of i vertices. This value is precisely p_i^Σ . This set of i vertices can be chosen in $\binom{k}{i}$ ways. Consequently, we want to estimate the sum $\sum_{i=0}^k \binom{k}{i} p_i^\Sigma$. Observe that this construction is quite close to Bell numbers, which count the number of ways a set of k elements can be partitioned into nonempty subsets. The main result of this section uses the following combinatorial trick :

Lemma 7.4 *Let $A(z) = \sum_{n>0} a_n z^n$. Then the sum $\sum_{i=0}^n \binom{n}{i} a_i$ is $[z^n] \frac{1}{1-z} A\left(\frac{z}{1-z}\right)$.*

Proof: It is a consequence of the Taylor development of $\frac{1}{1-z} A\left(\frac{z}{1-z}\right)$ and the relation $\frac{z^n}{(1-z)^{n+1}} = \sum_{i=0}^{\infty} \binom{n+i}{i} z^{n+i}$, which can be proved by induction. \square

Consequently, $\mathbf{P}_\Sigma(z, 1)$ is modified via Lemma 7.4 to obtain the GF for the numbers $\sum_{i=0}^k \binom{k}{i} p_i^\Sigma$. The singularity of $\mathbf{P}_\Sigma(z, 1)$ is located at $z = 1/4$, and therefore the singularity of $\frac{1}{1-z} \mathbf{P}_\Sigma(z/(1-z), 1)$ is located at $z = 1/5$. Its singular behavior (i.e., the singular exponent) is the same as the one for $\mathbf{P}_\Sigma(z, 1)$. The modification is made only on the position of the singularity, and not on its nature.

Summarizing, the estimate of $\sum_{i=0}^k \binom{k}{i} p_i^\Sigma$ for k big enough has exponential term equal to 5^k , and subexponential term equal to the one of p_k^Σ . In other words, we have proved the following proposition :

Proposition 7.7 *The following estimate holds :*

$$\frac{\sum_{i=0}^k \binom{k}{i} p_i^\Sigma}{p_k^\Sigma} =_{k \rightarrow \infty} \left(\frac{5}{4}\right)^{k+1} \left(1 + O(k^{-1/2})\right).$$

7.5.3 Putting the pieces together

Combining the univariate asymptotic obtained in Theorem 7.1 with the constructions described above (Propositions 7.6 and 7.7 in Sections 7.5.1 and 7.5.2, respectively) we obtain the following theorem, which gives the bound on the size of the tables when using surface cut decompositions :

Theorem 7.2 *Let $\overline{\Pi}_{\Sigma, l}(k)$ be the set of non-crossing partitions of Σ with k vertices and a set of l apices. Then the value $\sum_{i=0}^k \binom{k}{i} |\overline{\Pi}_{\Sigma, l}(k)|$ is upper-bounded, for large k , by*

$$\frac{C(\Sigma)}{2^{2+l} \Gamma(3/2\gamma(\Sigma) + \beta(\Sigma) - 3)} \cdot k^{3/2\gamma(\Sigma) + \beta(\Sigma) - 4 + l} \cdot 5^{k+1}, \quad (14)$$

where $C(\Sigma)$ is a function depending only on Σ that is bounded by $\gamma(\Sigma)^{O(\gamma(\Sigma))}$.

8 Conclusions and Open Problems

Our results can be summarized as follows.

Theorem 8.1 *Given a problem P belonging to Category (C) in a graph G embedded in a surface of Euler genus γ , with $\text{bw}(G) \leq k$, the size of the tables of a dynamic programming algorithm to solve P on a surface cut decomposition of G is bounded above by $2^{O(k)} \cdot k^{O(\gamma)} \cdot \gamma^{O(\gamma)}$.*

As we mentioned, the problems tackled in [11] are those in Category (B), which are included in Category (C). As a result of this, we reproduce all the results of [11]. Moreover, as our approach does not use planarization, our analysis provides algorithms where the dependence on the Euler genus γ is better than the one in [11]. In particular, the running time of the algorithms in [11] is $2^{O(\gamma \cdot \text{bw} + \gamma^2 \cdot \log(\text{bw}))} \cdot n$, while in our case the running time is $2^{O(\text{bw} + \gamma \cdot \log(\text{bw}) + \gamma \cdot \log \gamma)} \cdot n$.

Dynamic programming is important for the design of subexponential exact or parameterized algorithms. Using the fact that bounded-genus graphs have branchwidth at most $O(\sqrt{\gamma \cdot n})$ [18], we derive the existence of exact algorithms in $O^*(2^{O(\sqrt{\gamma \cdot n} + \gamma \cdot \log(\gamma \cdot n))})$ steps for all problems in Category (C). Moreover, using bidimensionality theory (see [9, 10]), one can derive $2^{O(\gamma \cdot \sqrt{k} + \gamma \cdot \log(\gamma \cdot k))} \cdot n^{O(1)}$ step parameterized algorithms for all bidimensional problems in Category (C).

Note that the running time of our algorithms is conditioned by the construction of an appropriate surface cut decomposition. This preprocessing step takes $2^{3k + O(\log k)} n^3$ steps by Theorem 6.1 of Section 6. Finding an alternative preprocessing algorithm with better polynomial dependence remains open.

A natural extension of our results is to consider more general classes of graphs than bounded-genus graphs. This has been done in [13] for problems in Category (B), where the tables of the algorithms encode pairings of the middle set. To extend these results for problems in Category (C) (where tables encode subsets of the middle set), using the planarization approach of [13], appears to be a quite complicated task. We believe that our surface-oriented approach could be more successful in this direction.

Notice that Categories (A), (B), and (C) can be seen as the first levels of a more general hierarchy of dynamic programming algorithms designed for gradually more complicated combinatorial problems. For instance, higher level classes of algorithms can be defined for tables encoding connected pairings (or even connected packings) of subsets of the middle set. In a sense, what we prove in this paper is the collapse of the time bounds in Category (C) to those in Category (A) when inputs are topologically restricted. It seems to be an interesting task to define such a hierarchy and to check whether this collapse extends to its higher levels.

Acknowledgements. We would like to thank Marc Noy for the idea of the proof of Lemma 7.4 of Section 7.5.2, and for pointing us to references [3, 21]. We would also like to thank Sergio Cabello for pointing us to the right topological references.

Références

- [1] E. Amir. Efficient approximation for triangulation of minimum treewidth. In *Proc. of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 7–15, 2001.
- [2] S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability – a survey. *BIT*, 25(1) :2–23, 1985.
- [3] E. A. Bender, Z. Gao, and L. B. Richmond. The map asymptotics constant t_g . *Electronic Journal of Combinatorics*, 15(1) :R51, 8pp, 2008.
- [4] O. Bernardi and J. Rué. Counting simplicial decompositions in surfaces with boundaries. Manuscript available at <http://lanl.arxiv.org/abs/0901.1608>.
- [5] H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Proc. of the 15th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 317 of *LNCS*, pages 105–118. Springer, Berlin, 1988.
- [6] S. Cabello and B. Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete and Computational Geometry*, 37 :213–235, 2007.
- [7] R. Cori. Indecomposable permutations, hypermaps and labeled dyck paths. *J. Comb. Theory Ser. A*, 116(8) :1326–1343, 2009.
- [8] B. Courcelle. The monadic second-order logic of graphs : definable sets of finite graphs. In *Proc. of the 14th International Workshop on Graph-theoretic Concepts in Computer Science (WG)*, volume 344 of *LNCS*, pages 30–53. 1988.
- [9] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H -minor-free graphs. *Journal of the ACM*, 52(6) :866–893, 2005.
- [10] E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos. The Bidimensional Theory of Bounded-Genus Graphs. *SIAM Journal on Discrete Mathematics*, 20(2) :357–371, 2006.
- [11] F. Dorn, F. V. Fomin, and D. M. Thilikos. Fast Subexponential Algorithm for Non-local Problems on Graphs of Bounded Genus. In *Proc. of the 10th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 4059 of *LNCS*, pages 172–183, 2006.
- [12] F. Dorn, F. V. Fomin, and D. M. Thilikos. Subexponential parameterized algorithms. In *Proc. of the 34th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4596 of *LNCS*, pages 15–27, 2007.
- [13] F. Dorn, F. V. Fomin, and D. M. Thilikos. Catalan structures and dynamic programming in H -minor-free graphs. In *Proc. of the 19th annual ACM-SIAM Symposium on Discrete Algorithms*

- (SODA), pages 631–640, 2008.
- [14] F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient Exact Algorithms on Planar Graphs : Exploiting Sphere Cut Branch Decompositions. In *Proc. of the 13th Annual European Symposium on Algorithms (ESA)*, volume 3669 of LNCS, pages 95–106, 2005.
 - [15] D. Eppstein. Dynamic Generators of Topologically Embedded Graphs. In *Proc. of the 14th annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 599–608, 2003.
 - [16] F. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge Univ. Press, 2008.
 - [17] P. Flajolet and M. Noy. Analytic combinatorics of non-crossing configurations. *Discr. Math.*, 204(1) :203–229, 1999.
 - [18] F. V. Fomin and D. M. Thilikos. Fast Parameterized Algorithms for Graphs on Surfaces : Linear Kernel and Exponential Speed-Up . In *Proc. of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3142 of LNCS, pages 581–592, 2004.
 - [19] F. V. Fomin and D. M. Thilikos. Dominating Sets in Planar Graphs : Branch-Width and Exponential Speed-Up. *SIAM Journal on Computing*, 36(2) :281–309, 2006.
 - [20] F. V. Fomin and D. M. Thilikos. On Self Duality of Pathwidth in Polyhedral Graph Embeddings. *Journal of Graph Theory*, 55(42-54), 2007.
 - [21] Z. Gao. The number of rooted triangular maps on a surface. *J. Combin. Theory Ser. B*, 52(2) :236–249, 1991.
 - [22] M. R. Henzinger, S. Rao, and H. N. Gabow. Computing Vertex Connectivity : New Bounds from Old Techniques. *Journal of Algorithms*, 34(2) :222–250, 2000.
 - [23] R. Impagliazzo, R. Paturi, and F. Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4) :512–530, 2001. Special issue of FOCS 1998.
 - [24] B. Mohar and C. Thomassen. *Graphs on surfaces*. John Hopkins University Press, 2001.
 - [25] N. Robertson and P. Seymour. Graph minors. X. Obstructions to Tree-decomposition. *J. Comb. Theory Series B*, 52(2) :153–190, 1991.
 - [26] N. Robertson and P. Seymour. Graph minors. XII. Distance on a surface. *J. Comb. Theory Series B*, 64 :240–272, 1995.
 - [27] I. Sau and D. M. Thilikos. Subexponential Parameterized Algorithms for Bounded-Degree Connected Subgraph Problems on Planar Graphs. *Electronic Notes in Discrete Mathematics*, 32 :59–66, 2009. Full version available at http://sites.thilikos.info/data/list/maxedge_full.pdf.

- [28] P. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2) :217–241, 1994.
- [29] J. A. Telle and A. Proskurowski. Algorithms for vertex partitioning problems on partial k -trees. *SIAM Journal on Discrete Mathematics*, 10(4) :529–550, 1997.
- [30] C. Thomassen. The graph genus problem is NP-complete. *Journal of Algorithms*, 10(4) :568–576, 1989.



Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399