Dynamic Programming for Graphs on Surfaces*

Juanjo Rué[1], Ignasi Sau[2], Dimitrios M. Thilikos[3]

[1] Laboratorie d'Informatique, École Polytechnique, 91128 Palaiseau-Cedex, France.
rue1982@lix.polytechnique.fr
[2] Department of Computer Science, Technion, Haifa, Israel.
ignasi@cs.technion.ac.il
[3] Dept. of Mathematics, National and Kapodistrian University of Athens, Greece.
sedthilk@math.uoa.gr

Abstract. We provide a framework for the design and analysis of dynamic programming algorithms for surface-embedded graphs on n vertices and branchwidth at most k. Our technique applies to general families of problems where standard dynamic programming runs in $2^{\mathcal{O}(k \cdot \log k)} \cdot n$ steps. Our approach combines tools from topological graph theory and analytic combinatorics. In particular, we introduce a new type of branch decomposition called *surface cut decomposition*, capturing how partial solutions can be arranged on a surface. Then we use *singularity analysis* over expressions obtained by the *symbolic method* to prove that partial solutions can be represented by a single-exponential (in the branchwidth k) number of configurations. This proves that, when applied on surface cut decompositions, dynamic programming runs in $2^{\mathcal{O}(k)} \cdot n$ steps. That way, we considerably extend the class of problems that can be solved in running times with a *single-exponential dependence* on branchwidth and unify/improve all previous results in this direction.

Keywords: analysis of algorithms; 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 dates back in the celebrated theorem of Courcelle (see e.g. [6]), stating that graph problems expressible in Monadic Second Order Logic can be solved in $f(\mathbf{bw}) \cdot n$ steps (here $\mathbf{bw}$ is the branchwidth[1] and n is the number of vertices of the input graph). Using pa-

* Research supported by the European Research Council under the EC's 7th Framework Programme, ERC grant agreement 208471 - ExploreMaps project, the Israel Science Foundation, grant No. 1249/08, and the project "Kapodistrias" (ΑΠ 02839/28.07.2008) of the National and Kapodistrian University of Athens.

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

rameterized complexity terminology, this implies that a large 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 a *single-exponential*– function, became a natural (and unavoidable) ingredient for many results on graph algorithms (see [3, 4, 10, 20]). In this paper, we provide a general framework for the design and analysis of dynamic programming algorithms for graphs embedded in surfaces where $f(\mathbf{bw}) = 2^{\mathcal{O}(\mathbf{bw})}$.

Dynamic programming. Dynamic programming is applied in a bottom-up fashion on a rooted branch decomposition of the input graph G, 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 S 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 to a middle set and the corresponding lower part. The size of these tables reflects the dependence on $k = |S|$ in the running time of the dynamic programming.

Designing the tables for each middle set S is not always an easy task and may vary considerably due to the particularities of each problem. The simplest cases are problems such as Vertex Cover and Dominating Set, where the certificate of the solution is a set of vertices whose choice is not restricted by some global condition. This directly yields to the desired $2^{\mathcal{O}(k)}$ upper bound on their size. For other problems, such as Longest Path, Cycle Packing, or Hamiltonian Cycle, things are more complicated as the tables encode *pairings of vertices of* S, which are $2^{\Theta(k \log k)}$ many. However, for such problems one can do better for *planar graphs* following the approach introduced in [12]. The idea in [12] is to use a special type of branch decomposition called *sphere cut decomposition* (introduced in [19]) that can guarantee that the pairings are *non-crossing* pairings 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 S by the k-th Catalan number, which is *single-exponential* in k. The same approach was extended for graphs embedded in surfaces of genus γ [9]. The idea in [9] 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 [12] to a more general version of the problem where the number of pairings is still bounded by some Catalan number (see also [11] for the application of this technique for more general graphs).

A wider family of problems are those where the tables of dynamic programming encode *packings* of S into sets; throughout this paper, we call these problems *packing-encodable problems*. Typical problems of this type are Connected Vertex Cover, Feedback Vertex Set, and Steiner Tree, where the connected components of a potential solution can be encoded by a collection of disjoint subsets of S, each of of *arbitrary cardinality*. Here, the general bound on the table size is given by the k-th Bell number, and thus again by $2^{\Theta(k \cdot \log k)}$. (To exemplify the differences between distinct encodings, typical dynamic programming algorithms for Vertex Cover and Connected Vertex Cover can be found in [18].) Unfortunately, for the latter category of problems, none of the current techniques is able to drop this bound to a single-exponential one for graphs embedded in surfaces.

Our results. In this paper, we follow a different approach in order to design single-exponential (in **bw**) algorithms for graphs embedded in surfaces. In particular, we deviate significantly from the planarization technique of [9], which is not able to tackle problems whose solutions are encoded by general packings. Instead, we extend the concept of sphere cut decomposition from planar graphs to graphs embeddable in generic surfaces, and we exploit directly the combinatorial structure of the potential solutions in the topological surface. Our approach permits us to provide in a unified way a single-exponential (in **bw**) time analysis for all aforementioned problems. Examples of other such problems are Connected Dominating Set, Connected r-Domination, Connected FVS, Maximum Leaf Spanning Tree, Maximum Full-Degree Spanning Tree, Maximum Eulerian Subgraph, or Maximum Leaf Tree. Our results imply all the results in [9, 12], and with running times whose genus dependence is better than the ones in [9], as discussed in Section 6.

Our techniques. For our results we enhance the current technology of dynamic programming with new tools for both topological graph theory and analytic combinatorics. We first propose a special type of branch decomposition of embedded graphs with nice topological properties, which we call *surface cut decomposition* (see Section 4). 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. Such a decomposition is based on the concept of *polyhedral decomposition* introduced in Section 3. We prove that the sizes of the tables of the dynamic programming correspond to the number of non-crossing partitions of vertex sets lying 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 [13]). 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 5). As this is the first time such a counting is done, our combinatorial results have independent mathematical interest.

For performing dynamic programming, our approach resides in a common preprocessing step that constructs a *surface cut decomposition* (Algorithm 2 in Section 4). 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 (Theorem 4 of Section 6). Due to space limitations, this extended abstract contains no proofs; they can be found in [18].

2 Preliminaries

Topological surfaces. In this paper, 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 [16] 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 $\overline{\Sigma}$ 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 $\chi(\overline{\Sigma}) = \beta(\Sigma) + \chi(\Sigma)$. A subset Π of a surface Σ is *surface-separating* if $\Sigma \setminus \Pi$ has at least 2 connected components. It is convenient to work with the *Euler genus* $\gamma(\Sigma)$ of a surface Σ, which is defined as $\gamma(\Sigma) = 2 - \chi(\Sigma)$.

Graphs embedded in surfaces. 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. In a *2-cell embedding*, also called *map*, each face is homeomorphic to a disk. The degree $\mathrm{d}(v)$ 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.

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. 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* [17, 19], 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* [16] 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 by 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 $\{u^*, v\}$. 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.

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 $\mu : L \to E(G)$ 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., $\mathbf{w}(T, \mu) := \max\{|\mathbf{mid}(e)| \colon e \in T\}$. 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 [19].

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.

3 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 [5], we use the notation $G {\times\!\!\!\times} N$ for the graph obtained by cutting G along the noose N and gluing a disk on the obtained boundaries.

Definition 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 $\mathcal{G} = \{H_1, \ldots, H_\ell\}$ together with a set of vertices $A \subseteq V$ such that*

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

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

Algorithm 1 Construction of a polyhedral decomposition of an embedded graph

---

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 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 $H[V(H) \setminus S] \mathbin{\succ\!\!\!\prec} N$.
  if N is surface-separating **then**
    Let H_1, H_2 be the subgraphs of $H \mathbin{\succ\!\!\!\prec} N$ 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 $H \mathbin{\succ\!\!\!\prec} N$ containing at least one edge of H.
return $\{\mathcal{G}, A\}$.

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

4 Surface Cut Decompositions

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 2. *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 $|A_e| = \mathcal{O}(\gamma(\Sigma))$ and either*

- $|\mathbf{mid}(e) \setminus A_e| \leq 2$, *or*
- *there exists a polyhedral decomposition $\{\mathcal{G}, A\}$ 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 $\mathcal{N}$ of $\mathcal{O}(\gamma(\Sigma))$ nooses, such that the total number of occurrences in $\mathcal{N}$ of the vertices in $\mathbf{mid}(e) \setminus A_e$ is $|\mathbf{mid}(e) \setminus A_e| + \mathcal{O}(\gamma(\Sigma))$; 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 now show in Algorithm 2 how to construct a surface cut decomposition of an embedded graph. More details can be found in [18].

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

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 [2, 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.
3. Transform (T^c_H, μ^c_H) to a *bond* carving decomposition (T^b_H, μ^b_H) of M_H, using [19].
4. Transform (T^b_H, μ^b_H) to a branch decomposition (T_H, μ_H) of H.

Construct a branch decomposition (T, μ) of G by merging 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 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 $2^{3k+\mathcal{O}(\log k)} \cdot n^3$ steps, a surface cut decomposition (T, μ) of G of width at most $27k + \mathcal{O}(\gamma)$.*

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 packing-encodable problems. The size of the tables depends on how many ways a partial solution can intersect a middle set during the dynamic programming algorithm. The advantage 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 $\mathcal{O}(\gamma)$ nooses, where vertices can be repeated at most $\mathcal{O}(\gamma)$ 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 2). 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 $\mathcal{O}(\gamma)$ disjoint nooses containing ℓ vertices, which are $2^{\mathcal{O}(\ell)} \cdot \ell^{\mathcal{O}(\gamma)} \cdot \gamma^{\mathcal{O}(\gamma)}$, as we prove in the following section (Theorem 3). Observe that the splitting operation increases the size of the middle sets by at most $\mathcal{O}(\gamma)$, therefore $\ell = k + \mathcal{O}(\gamma)$ and this yields an upper bound of $2^{\mathcal{O}(k)} \cdot k^{\mathcal{O}(\gamma)} \cdot \gamma^{\mathcal{O}(\gamma)}$ on the size of the tables of the dynamic programming. In Section 5 we use singularity analysis over expressions obtained by the symbolic method to count the number of such non-crossing partitions. Namely, in Sections 5.1 and 5.2 we give a precise estimate of the number of non-crossing partitions in surfaces with boundary. Then we incorporate 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 5 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. [14]).

5 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 Section 5.1 we set up our notation. In Section 5.2 we obtain a tree-like structure that provides a way to obtain asymptotic estimates. In this part, we exploit map enumeration techniques, together with singularity analysis.

5.1 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 of this section is to obtain 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, the following propositionshows that it is sufficient to study regular decompositions:

Proposition 2. *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 $\pi_{\Sigma}(s) = \pi_{\Sigma}(m)$.*

In other words, $|\Pi_{\Sigma}(k)| \leq |\mathcal{S}_{\Sigma}(k)| \leq |\mathcal{R}_{\Sigma}(k)|$ 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 that, as we show later, 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 can be found in [18].

5.2 Tree-like structures, enumeration, and asymptotic counting

In this subsection we provide estimates for the number of irreducible 2-zone decompositions, which are obtained directly for the surface Σ. The main point consists in exploiting tree-like structures of the dual graph associated to an irreducible 2-zone decomposition. 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). We use this family of trees in order to obtain a decomposition of elements of the set $\mathcal{P}_{\Sigma}(k)$. (The reader which is not familiar with the symbolic method and analytic combinatorics is referred to [13].) In [18] the enumeration of this basic family is done, as well as the enumeration of the related families.

The construction for general surfaces is a generalization of the previous one. An example is shown in the leftmost picture of Fig. 1. For an element $m \in \mathcal{P}_{\Sigma}(k)$, denote by M the resulting map on $\overline{\Sigma}$ (recall the definition of $\overline{\Sigma}$ 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 $\overline{\Sigma}$ as follows: 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 Fig. 1 for an example.

An inverse construction can be done using maps over $\overline{\Sigma}$ and families of plane trees. Using these basic pieces, we can reconstruct all irreducible 2-zone decompositions. Exploiting this decomposition and using singularity analysis, we get the following theorem (Γ denotes the classical Gamma function [13]):

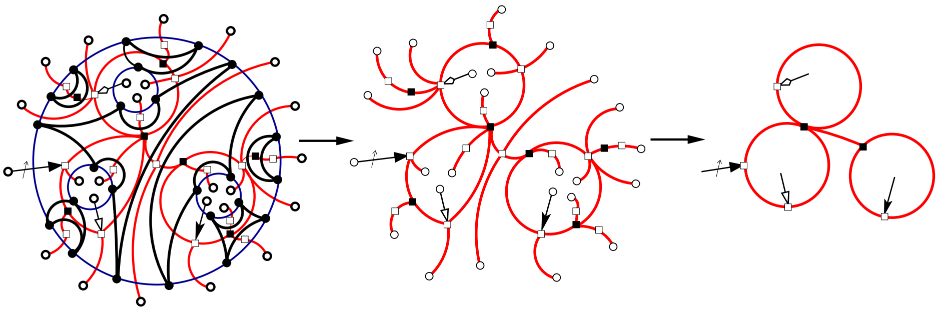

Fig. 1. Construction of the scheme of an element in $\mathcal{P}_\Sigma$. The dual of an irreducible 2-zone decomposition is shown on the left. After deleting vertices of degree 1 recursively and dissolving vertices of degree 2, we obtain the associated scheme on the right.

Theorem 2. *Let Σ be a surface with boundary. Then,*

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

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

Additional constructions. So far, we enumerated families of non-crossing partitions with boundary. Firstly, in surface cut decompositions we need to deal with a set of additional vertices that play the role of *apices* (cf. the last paragraph of Section 4). Secondly, we show how to extend the enumeration from non-crossing partitions to non-crossing packings. In both cases, we show that the modification over generating functions (GFs for short) does 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. Combining the univariate asymptotic obtained in Theorem 2 with the constructions described above, we obtain the bound on the size of the tables when using surface cut decompositions:

Theorem 3. *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 $\sum_{i=0}^{k}\binom{k}{i}\left|\overline{\Pi_{\Sigma,l}}(k)\right|$ is upper-bounded, for large k, by*

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

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

6 Conclusions and Open Problems

Our results can be summarized as follows.

Theorem 4. *Given a packing-encodable problem P in a graph G embedded in a surface of Euler genus γ, with $\mathbf{bw}(G) \leq k$, the size of the tables of a dynamic programming algorithm to solve P on a surface cut decomposition of G is bounded above by $2^{\mathcal{O}(k)} \cdot k^{\mathcal{O}(\gamma)} \cdot \gamma^{\mathcal{O}(\gamma)}$.*

As we mentioned, the problems tackled in [9] can be encoded with pairings, and therefore they can be seen as special cases of packing-encodable problems. As a result of this, we reproduce all the results of [9]. 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 [9]. In particular, the running time of the algorithms in [9] is $2^{\mathcal{O}(\gamma \cdot \mathbf{bw} + \gamma^2 \cdot \log(\mathbf{bw}))} \cdot n$, while in our case the running time is $2^{\mathcal{O}(\mathbf{bw} + \gamma \cdot \log(\mathbf{bw}) + \gamma \cdot \log \gamma)} \cdot n$.

Dynamic programming is important for the design of *subexponential* exact or parameterized algorithms. Using the fact that bounded-genus graphs have branchwidth at most $\mathcal{O}(\sqrt{\gamma \cdot n})$ [15], we derive the existence of exact algorithms in $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{\gamma n} + \gamma \cdot \log(\gamma \cdot n))})$ steps for all packing-encodable problems. Moreover, using bidimensionality theory (see [7,8]), one can derive $2^{\mathcal{O}(\gamma \cdot \sqrt{k} + \gamma \cdot \log(\gamma \cdot k))} \cdot n^{\mathcal{O}(1)}$ step parameterized algorithms for all bidimensional packing-encodable problems.

Sometimes dynamic programming demands even more complicated encodings. We believe that our results can also serve in this direction. For instance, surface cut decompositions have recently been used in [1] for minor containment problems, where tables encode partitions of packings of the middle sets.

A natural extension of our results is to consider more general classes of graphs than bounded-genus graphs. This has been done in [11] for problems where the tables of the algorithms encode pairings of the middle sets. To extend these results for packing-encodable problems (where tables encode subsets of the middle sets) using the planarization approach of [11] appears to be a quite complicated task. We believe that our surface-oriented approach could be more successful in this direction and we find it an interesting, but non-trivial task.

Acknowledgement. We would like to thank Marc Noy and Sergio Cabello for valuable ideas and for pointing us to several interesting references.

References

1. I. Adler, F. Dorn, F. V. Fomin, I. Sau, and D. M. Thilikos. Faster Parameterized Algorithms for Minor Containment. In *Proc. of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, 2010. To appear.
2. E. Amir. Efficient approximation for triangulation of minimum treewidth. In *Proc. of the 17th Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 7–15, 2001.
3. S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability – a survey. *BIT*, 25(1):2–23, 1985.
4. H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Proc. of the 15th International Colloquium on Automata, Languages and Programming (ICALP)*, LNCS 317, pages 105–118. 1988.

5. 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.
6. 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)*, LNCS 344, pages 30–53. 1988.
7. 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.
8. 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.
9. 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)*, LNCS 4059, pages 172–183, 2006.
10. 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)*, LNCS 4596, pages 15–27, 2007.
11. 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.
12. 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)*, LNCS 3669, pages 95–106, 2005.
13. F. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2008.
14. P. Flajolet and M. Noy. Analytic combinatorics of non-crossing configurations. *Discrete Mathematics*, 204(1):203–229, 1999.
15. 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)*, LNCS 3142, pages 581–592, 2004.
16. B. Mohar and C. Thomassen. *Graphs on surfaces.* John Hopk. Univ. Press, 2001.
17. N. Robertson and P. Seymour. Graph minors. XII. Distance on a surface. *J. Combin. Theory Series B*, 64:240–272, 1995.
18. J. Rué, I. Sau, and D. M. Thilikos. Dynamic Programming for Graphs on Surfaces. Research Report RR-7166, INRIA, 2009. Available at `hal.archives-ouvertes.fr/inria-00443582`.
19. P. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.
20. 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.