18th Annual International Computing and Combinatorics Conference, COCOON 2012, August 20–22, 2012.

Dynamic Programming for H-minor-free Graphs*

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

[1] Instituto de Ciencias Matemáticas, Madrid, Spain.
juanjo.rue@icmat.es
[2] AlGCo project-team, CNRS, LIRMM, Montpellier, France.
ignasi.sau@lirmm.fr
[3] Department 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 H-minor-free graphs with branchwidth at most k. Our technique applies to a wide family of problems where standard (deterministic) dynamic programming runs in $2^{O(k \cdot \log k)} \cdot n^{O(1)}$ steps, with n being the number of vertices of the input graph. Extending the approach developed by the same authors for graphs embedded in surfaces, we introduce a new type of branch decomposition for H-minor-free graphs, called an *H-minor-free cut decomposition*, and we show that they can be constructed in $O_h(n^3)$ steps, where the hidden constant depends exclusively on H. We show that the separators of such decompositions have connected packings whose behavior can be described in terms of a combinatorial object called *ℓ-triangulation*. Our main result is that when applied on H-minor-free cut decompositions, dynamic programming runs in $2^{O_h(k)} \cdot n^{O(1)}$ steps. This broadens substantially the class of problems that can be solved deterministically in *single-exponential* time for H-minor-free graphs.

Keywords: analysis of algorithms; parameterized algorithms; graphs minors; branchwidth; dynamic programming; non-crossing partitions.

1 Introduction

The celebrated theorem of Courcelle [7] states that graph problems expressible in MSOL can be solved in $f(\mathbf{bw}) \cdot n$ steps, where $\mathbf{bw}$ is the branchwidth and n is the number of vertices of the input graph. Using terminology from parameterized complexity, this implies that a large number of graph problems admit fixed-parameter tractable algorithms 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 specific dynamic programming algorithms for graph problems so that $f(\mathbf{bw})$ is a simple function, became an essential ingredient for many results on graph algorithms (see [2, 4, 12, 13, 33]). In this paper, we provide a general framework for the design and analysis of dynamic programming algorithms for families of graphs excluding a graph H as a minor, where $f(\mathbf{bw}) = 2^{O(\mathbf{bw})}$.

* This research was done during a research visit of the first two authors at the Department of Mathematics of the National and Kapodistrian University of Athens. The authors wish to express their gratitude to the decisive support of "Pontios" during that visit.

** Partially supported by grants JAE-DOC (CSIC), MTM2011-22851, and SEV-2011-0087.

*** Partially supported by project AGAPE (ANR-09-BLAN-0159) from France.

† Co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: "Thales. Investing in knowledge society through the European Social Fund".

Our framework applies to a family of problems where no deterministic dynamic programming algorithm with single-exponential parameterized dependence on **bw** is known.

Motivation and previous work. Dynamic programming is usually applied in a bottom-up fashion on a rooted branch decomposition of the input graph G. Roughly, a branch decomposition of a graph is a way to decompose it into a tree structure of edge bipartitions (the formal definition is in Section 2). Each bipartition defines a separator S of the graph called the *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 a 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 may vary considerably among different problems. For simple problems where the tables of dynamic programming encode vertex subsets of the middle set, such as Vertex Cover or Dominating Set, we may easily have a single-exponential dependence on k, as the number of subsets of a set of size k is 2^k. However, there are problems where the tables of the dynamic programming encode vertex pairings, such as Longest Path, Cycle Packing, or Hamiltonian Cycle, or (more generally) vertex packings, such as Connected Vertex Cover, Connected Dominating Set, Feedback Vertex Set, or Steiner Tree. For the latter category of problems, single-exponential bounds on their table size are not known to exist. This complication arises from the fact that, for such problems, the tables should encode at least $2^{\Theta(k \log k)}$ many pairings/packings. Nevertheless, for such problems one can do better for several classes of sparse graphs. This line of research was initiated in [14] and occupied several researchers in parameterized algorithms design (see also [5,8,11,13,24,33]). The current technology of dynamic programming in graphs of bounded decomposability implies single-exponential parametric dependence for problems encodable by pairings in H-minor free graphs [13] and for problems encodable by packings in graphs embedded in surfaces [33]. In this paper we extend both approaches of [13] and [33] to problems encodable by packings in H-minor free graphs.

Our results and techniques. We present a general framework that provides single-exponential dynamic programming algorithms for *connected packing-encodable problems* (the formal definition of this class of problems is in Section 3) when the input graph excludes a graph H as a minor. The main idea in [33] was to introduce a new type of branch decomposition for graphs on surfaces, called a *surface cut decomposition* (which in turn, extended the concept of sphere cut decompositions for planar graphs introduced in [14,34]). Namely, in [33], it was proved that the number of partial solutions that can be arranged on a surface cut decomposition can be upper-bounded by the number of non-crossing partitions on surfaces with boundary, which have been recently enumerated in [32]. It follows that partial solutions can be represented by a single-exponential number of configurations. This proves that, when applied on surface cut decompositions, dynamic programming for connected packing-encodable problems runs in $2^{O(k)} \cdot n^{O(1)}$ steps.

We follow the same approach to extend this technique to H-minor-free graphs: we define a new type of branch decomposition for graphs excluding an h-vertex graph H as a minor; we call it an *H-minor-free cut decomposition*. In Section 5 we show how to compute an H-minor-free cut decomposition of width $O_h(k)$ in $O_h(n^3)$ steps[1]. This

[1] Given a computable function f and an integer h, we use the notation $O_h(f(k))$ to denote $O(g(h) \cdot f(k))$ for some computable function g.

algorithm uses the recent result of Kawarabayashi and Wollan [21] to find in time $O(n^3)$ the tree-like decomposition of an H-minor-free graph G, given by the seminal structure theorem of Robertson and Seymour [30]. Roughly, this result says that each H-minor free graph admits a bounded-adhesion tree decomposition whose bags are nearly embedded in some surface of small genus. We also make use of the algorithm of [33] to find a surface cut decomposition of the surface-embedded part of each bag, then enhance them with the apices and vortices, and finally we glue them appropriately along the clique-sums. But for being able to use the algorithm of [33], we need to prove that there exists a tree-like decomposition of G whose bags have good topological properties. This is done in Section 4, and requires a suitable extension of the notion of a polyhedral decomposition introduced in [33].

In order to prove the upper bound on the size of the tables when using an H-minor-free cut decomposition, the main difficulty is to deal with the vortices. From a combinatorial point of view, our main contribution is to capture the behavior of the vortices of an H-minor-free graph in terms of an object called *ℓ-triangulation* (cf. Section 2). Roughly speaking, in order to take into account the number of *simultaneous* crossings of a set of connected subgraphs inside a vortex, ℓ-triangulations seem to be the appropriate combinatorial object to look at (see Section 6 for more details). Finally, we prove our main result in Section 7. That is, by combining all the ingredients mentioned above, we prove that by using H-minor-free cut decompositions, the size of the tables for solving connected packing-encodable problems is single-exponential in the branchwidth. We would like to note that we did not make any effort to optimize the constants depending on H, as they are already huge since we use the Structure Theorem of the Graph Minors series [21, 30].

Our results can also be used to derive subexponential parameterized algorithms for connected packing-encodable *bidimensional* problems. That way, we broaden the class of problems where the general framework introduced in [9] can be applied. It is worth mentioning that our results directly imply that Steiner Tree and Connected Dominating Set, among others, can be solved in subexponential time in H-minor-free graphs, which has been recently (and independently) proved by Tazari [35].

Recent results and further research. Recently, Cygan *et al.* [8] have presented a new framework for obtaining *randomized* single-exponential algorithms parameterized by treewidth in general graphs. This framework is based on a dynamic programming technique named Cut&Count, which seems applicable to most connected packing-encodable problems, like Connected Vertex Cover, Connected Dominating Set, Feedback Vertex Set, or Steiner Tree. The randomization in the algorithms of [8] comes from the usage a probabilistic result called the Isolation Lemma [26], whose derandomization is a challenging open problem [3]. Therefore, the existence of *deterministic* single-exponential algorithms parameterized by treewidth for connected packing-encodable problems in general graphs remains wide open.

Our results for minor-free graphs can be seen as an intermediate step towards an eventual positive answer to this question. It may also be the case that this class of graphs establishes a frontier of the existence of *deterministic* single-exponential parameterized algorithms for connected packing-encodable problems (one way to prove this would be to use the recent approach given by Lokshtanov *et al.* [24] in order to provide lower bounds of running times of problems parameterized by treewidth), although we do not think that this is the case. It would be also interesting to reduce the big polynomial overheads in the algorithms of [8], given by the usage of the Isolation Lemma. In addition, the approach presented in [8] does not seem to be applicable to weighted problems, while our results are easily extendable to weighted connected packing-encodable problems.

If one wants to further extend our approach to more general classes of graphs, the natural candidate are graphs excluding a graph H as a topological minor, possibly using the recent structural results of Grohe and Marx [18]. Nevertheless, our framework seems hard to extend to graphs of bounded degree, which are topological-minor-free.

Finally, it is worth mentioning that another type of branch decomposition for graphs on surfaces, called *s*urface split decomposition, has been recently introduced by Bonsma [5] to prove that Subgraph Isomorphism can be solved in single-exponential time in graphs on surfaces. It remains open to find single-exponential algorithms for Subgraph Isomorphism in H-minor-free graphs.

2 Preliminaries

All graphs we consider are finite, simple, and undirected. Given a graph G and an edge $e \in E(G)$, let G/e be the graph obtained from G by contracting e, removing loops and parallel edges. If H can be obtained from a subgraph of G by a (possibly empty) sequence of edge contractions, we say that H is a *minor* of G. In this paper, we consider graphs embedded in surfaces that are compact and their boundary is homeomorphic to a (possibly empty) finite set of disjoint circles. We denote by $\nu(\Sigma)$ the number of connected components of the boundary of a surface Σ. 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 (see [25] for the precise definitions). A subset Π of a surface Σ is *surface-separating* if $\Sigma \setminus \Pi$ has at least two connected components.

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 defined as $\mathbf{w}(T, \mu) := \max\{|\mathbf{mid}(e)| \mid e \in T\}$. 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 $\mathbf{bw}(G)$. By definition (see [29]), the branchwidth of a graph G with $|E(G)| \leq 1$ is taken to be zero.

A *tree decomposition* of a graph G is a pair $\mathcal{D} = (\mathcal{X}, T)$ where T is a tree and $\mathcal{X} = \{X^t \mid t \in V(T)\}$ is a collection of subgraphs of $V(G)$ such that: $\bigcup_{t\in V(T)} X^t = G$ and for each $x \in V(G)$, the set $\{t \mid x \in V(X^t)\}$ induces a connected subtree of T. We call the vertices of T *nodes* of $\mathcal{D}$ and we call the graphs in $\mathcal{X}$ *bags* of $\mathcal{D}$. The *width* of a tree decomposition $\mathcal{D}$ is $\max\{|V(X^t)| - 1 \mid t \in V(T)\}$ and its *adhesion* is $\max\{|V(X^t) \cap V(X^{t'})| \mid t, t' \in V(T)\}$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G. For every $X^t, t \in V(T)$ we denote by $\mathcal{C}(X^t) = \{V(X^t) \cap V(X^{t'}) \mid t' \in N_T(t)\}$ and we define the $\mathcal{D}$*-closure* of X^t, as the graph $\overline{X}^t = X^t \cup \left(\bigcup_{C\in\mathcal{C}(X^t)} K[C]\right)$, where $K[C]$ is the clique obtained if we connect all pairs of distinct vertices of C. We finally define $\overline{G} = \bigcup_{t\in V(T)} \overline{X}^t$.

ℓ-triangulations and related constructions. Let $\mathbb{D}_k$ be a disc with k vertices on its border. We assume that these vertices are labeled counterclockwise with labels $1, 2, \ldots, k$. By an ℓ*-triangulation* of $\mathbb{D}_k$ we mean a maximal set of diagonals with no pairwise crossing-set of size $\ell + 1$. In other words, the graph whose vertices are the diagonals of the ℓ-triangulation and there is an edge between two diagonals if and only if they cross in an internal vertex, does not contain $K_{\ell+1}$ as a subgraph. This concept generalizes the classical notion of *triangulation* of a disc, as 1-triangulations correspond to triangulations. Denote by $T_\ell(k)$ the number of different ℓ-triangulations of $\mathbb{D}_k$. In particular, $T_0(k+2)$ is equal to the k-th Catalan number $C_k = \frac{1}{k+1}\binom{2k}{k}$, a result which is well-known since Euler's time. The study of ℓ-triangulations for $\ell > 1$ is more involved than the study of triangulations. In [16, 27] the authors show that the number of diagonals in an ℓ-triangulation of $\mathbb{D}_k$ is always $\ell(k - 2\ell - 1)$. More recently, in [20] the following closed expression for $T_\ell(k)$ has been obtained:

$$T_\ell(k) = \begin{vmatrix} C_{k-2} & C_{k-3} & \cdots & C_{k-\ell-1} \\ C_{k-3} & C_{k-4} & \cdots & C_{k-\ell-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{k-\ell-1} & C_{k-\ell-2} & \cdots & C_{k-2\ell} \end{vmatrix}. \tag{1}$$

This expression generalizes the enumeration of the number of triangulations of a polygon with k vertices. For the asymptotic of $T_\ell(k)$, observe that the recurrence $C_k = \frac{4k-2}{k+1}C_{k-1}$ makes each entry of the determinant equal to C_k times a rational function of degree at most 2ℓ in ℓ. Using also that $C_k = \frac{1}{\sqrt{\pi}}k^{-3/2}4^k(1 + o(1))$ for k large enough, it is easy to get bounds for $T_\ell(k)$. More concretely, $T_\ell(k) \leq_{k\to\infty} \frac{\ell!}{\pi^{\ell/2}}k^{-3\ell/2}4^{\ell k}$.

We say that a set of diagonals in $\mathbb{D}_k$ is a *partial ℓ-triangulation* if it a subset of an ℓ-triangulation. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_r\}$ be a partition of the set $\{1, 2, \ldots, k\}$ (i.e., $\bigcup_{i=1}^r \pi_i = \{1, 2, \ldots, r\}$, and $\pi_i \cap \pi_j = \emptyset$ if and only if $i \neq j$). We say that each of the subsets π_i, $i \in \{1, 2, \ldots, r\}$ is a *block* of the partition Π. We represent a partition in the following way: we draw each block of Π as a polygon connecting the corresponding vertices. This defines a graph $G(\Pi)$ whose vertices are the blocks of Π, and the edges are defined by the incidences of the blocks (i.e., an edge is drawn between a block π_i and a block π_j if and only if the associated polygons intersect in the graphical representation). We say that a packing (that is, a collection of pairwise disjoint non-empty blocks) of the disc $\mathbb{D}_k$ is an *ℓ-packing* if and only if $G(\Pi)$ does not contain $K_{\ell+1}$ as a subgraph. In particular, if $\ell < \ell'$, then an ℓ-packing is also an ℓ'-packing. The notion of ℓ-packing of a disc is a natural generalization of the notion of non-crossing partition, which corresponds to the case $\ell = 1$, in the same way as ℓ-triangulations generalize triangulations of a disc. In the leftmost side of Fig. 1 a 3-packing is drawn. In the following lemma we find asymptotic estimates for the number of ℓ-packings of $\mathbb{D}_k$, which we denote by $P_\ell(k)$.

Lemma 1. *The number of ℓ-packings of $\mathbb{D}_k$ satisfies $P_\ell(k) = 2^{O_\ell(k)}$.*

Proof: Our aim is to prove that the number of ℓ-packings of $\mathbb{D}_k$ satisfies

$$P_\ell(k) \;\leq\; 2^{\ell(k-2\ell-1)} \cdot T_\ell(k) \;\leq_{k\to\infty}\; \frac{\ell!}{2^{2\ell^2+\ell}\cdot\pi^{\ell/2}} \cdot k^{-3\ell/2} \cdot 8^{h\ell k} \;=\; 2^{O_\ell(k)}.$$

We construct an injective application from ℓ-packings of $\mathbb{D}_k$ into partial ℓ-triangulations of $\mathbb{D}_k$. For each block, we consider the first vertex we meet when we move around $\mathbb{D}_k$ starting at vertex 1. From each one of these vertices we draw diagonals to the rest of the vertices of the block. Then it is obvious that the resulting set of diagonals is a partial ℓ-triangulation: a pair of diagonals coming from the same block do not cross, and for each pair of crossing blocks there exists at least a pair of crossing diagonals. See the rightmost side of Fig. 1 for an explicit construction. As each partial ℓ-triangulation is obtained from a (maximal) ℓ-triangulation by deleting a subset of edges, and each (maximal) ℓ-triangulation has $\ell(k - 2\ell - 1)$ edges, the result follows. □

Partitions of an integer. Let q be a non-negative integer. A *partition* of q is a non-increasing sequence of positive integers $p_1, p_2, \ldots, p_r$ whose sum is q. We denote by $p(q)$ the number of partitions of q. The Hardy-Ramanujan-Rademacher estimate for $p(q)$ (see [1] for details) states that $p(q) = \frac{1}{4\sqrt{3}q}e^{\pi\sqrt{2q/3}}(1 + o(1)) = 2^{O(\sqrt{q})}$.

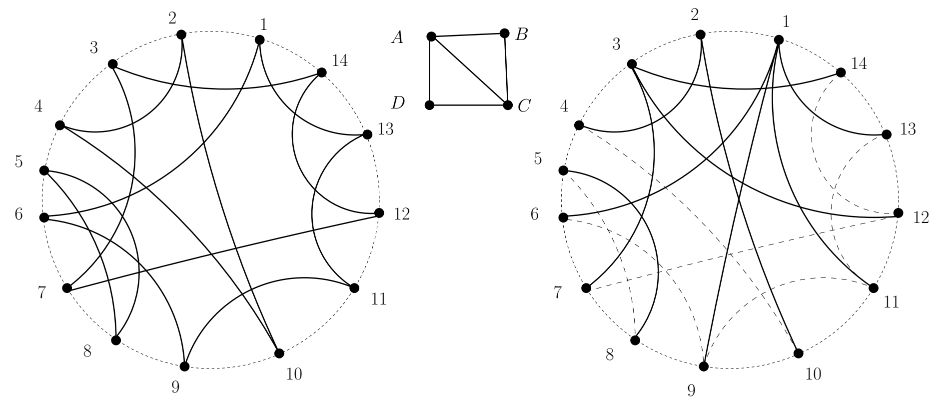

Fig. 1. A 3-packing of the disc $\mathbb{D}_{14}$ with blocks $A = \{1, 6, 9, 11, 13\}$, $B = \{2, 4, 10\}$, $C = \{3, 7, 12, 14\}$ and $D = \{5, 8\}$. On the left, the incidence graph of the partition, and on the right, the associated partial 3-triangulation (used in the proof of Lemma 1).

3 Connected packing-encodable problems

The standard dynamic programming approach on branch decompositions requires a so called *rooted* branch decomposition, defined as a triple (T, μ, e_r), where (T, μ) is a branch decomposition of G such that T is a tree rooted at a leaf v_r of T incident with some edge e_r. We slightly abuse notation by insisting that no edge of G is assigned to v_r and thus $\mathbf{mid}(e_r) = \emptyset$ (for this, we arbitrarily pick some edge of a branch decomposition, subdivide it and then connect by e_r the subdivision vertex with a new leaf v_r). The edges of T are 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 with leaves of T. Given an edge e whose tail is a non-leaf vertex v, we denote by $e_1, e_2 \in E(T)$ the two edges heading at v (we call them *children* of e). When the tail of an edge of T is also a leaf of T then we call it *leaf-edge*.

Typically, dynamic programming on a rooted branch decomposition (T, μ, e_r) of a graph G associates some suitable combinatorial structure $\mathsf{struct}(e)$ with each edge e of T, such that the knowledge of $\mathsf{struct}(e_r)$ makes it possible to determine the solution to the problem. Roughly speaking, $\mathsf{struct}(e)$ encodes all the ways that the possible certificates of a partial solution on graph G_e may be restricted to $\mathbf{mid}(e)$. The computation of $\mathsf{struct}(e)$ is done bottom-up, by first providing $\mathsf{struct}(e)$ when e is a leaf-edge of T and then giving a recursive way to construct $\mathsf{struct}(e)$ from $\mathsf{struct}(e_1)$ and $\mathsf{struct}(e_2)$, where e_1 and e_2 are the children of e.

The encoding of struct is commonly referred as the "tables" of the dynamic programming algorithm. It is desirable that the size of the tables, as well as the time to process them, is bounded by $f(|\mathbf{mid}(e)|) \cdot n^{O(1)}$, where f is a function not depending on n. This would give a polynomial-time algorithm for graphs of fixed branchwidth. In technical terms, this means that the problem is *Fixed Parameter Tractable* (FPT), when parameterized by the branchwidth of the input graph (for more on Fixed Parameter Tractability, see [15, 17, 28]). A challenge in the design of such algorithms is to reduce the contribution of branchwidth to the size of their tables, and therefore to simplify f as much as possible. As indicated by the lower bounds in [6, 19, 23], for many problems like Independent Set, Dominating Set, or q-Coloring for fixed $q \geq 3$, f is not expected to be better than single-exponential in general graphs.

Before we proceed with the description of the family of problems that we examine in this paper, we need some definitions. Let G be a graph and let S be a set of vertices of G. We denote by $\mathcal{G}$ the collection of all subgraphs of G. Each $H \in \mathcal{G}$ defines a packing $\mathcal{P}_S(H)$ of S such that two vertices $x, y \in S$ belong to the same set of $\mathcal{P}_S(H)$ if x, y belong to the same connected component of H. We say that $H_1, H_2 \in \mathcal{G}$ are *S-equivalent* if $\mathcal{P}_S(H_1) = \mathcal{P}_S(H_2)$, and we denote it by $H_1 \equiv_S H_2$. Let $\overline{\mathcal{G}}_S$ the collection of all subgraphs of G modulo the equivalence relation $\equiv_S$. We define the set of all *connected packings of S with respect to G* as the collection $\Psi_G(S) = \{\mathcal{P}_S(H) \mid H \in \overline{\mathcal{G}}_S\}$. Notice that each member of $\Psi_G(S)$ can indeed be seen as a packing of S, as its sets may not necessarily meet all vertices of S.

In this paper we consider graph problems that can be solved by dynamic programming algorithms on branch decompositions for which the size of $\mathsf{struct}(e)$ is upper-bounded[2] by $2^{O(|\mathbf{mid}(e)|)} \cdot |\Psi_{G_e}(\mathbf{mid}(e))| \cdot n^{O(1)}$. We call these problems *connected packing-encodable.* We stress that our definition of connected packing-encodable problem assumes the existence of *an* algorithm with this property, but there may exist other algorithms whose tables are much bigger. For these problems, dynamic programming has a single-exponential dependence on branchwidth if and only if $\Psi_{G_e}(\mathbf{mid}(e))$ contains a single-exponential number of packings, i.e., $|\Psi_{G_e}(\mathbf{mid}(e))| = 2^{O(|\mathbf{mid}(e)|)}$. See [33] for more details.

In general, the number of different connected packings that could be created during the dynamic programming is not necessarily smaller than the number of the non-connected ones. In fact, it typically depends on the k-th Bell number, where k is the branchwidth of the input graph. This implies that, in general, $|\Psi_{G_e}(\mathbf{mid}(e))| = 2^{O(k \log k)}$ is the best upper bound that can be so far achieved for connected packing-encodable problems, at least for deterministic algorithms. The purpose of this paper is to show that, for such problems, this bound can be reduced to a single-exponential one when their input graphs exclude a graph as a minor. In Section 5, we define the concept of an H-minor-free cut decomposition, which is a key tool for the main result of this paper, summarized as follows.

Theorem 1. *Every connected packing-encodable problem whose input is an n-vertex graph G that excludes an h-vertex graph H as a minor, and has branchwidth at most k, can be solved by a dynamic programming algorithm on an H-minor-free cut decomposition of G with tables of size $2^{O_h(k)} \cdot n^{O(1)}$.*

In Section 3, we prove (Theorem 3) that, given an H-minor-free graph G, an H-minor-free cut decomposition of G of width $O_h(\mathbf{bw}(G))$ can be constructed in $O_h(n^3)$ steps. Therefore, we conclude the following result.

Theorem 2. *Every connected packing-encodable problem whose input is an n-vertex graph G that excludes an h-vertex graph H as a minor and has branchwidth at most k, can be solved in $2^{O_h(k)} \cdot n^{O(1)}$ steps.*

4 Polyhedral decomposition of H-minor-free graphs

Let Σ be a surface with boundary. 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

[2] In most cases this bound is independent from n. We choose to present it in a more general form that might include graph problems that are not covered by the linear time algorithms derived from Courcelles's theorem.

G-normal, then we call it a *noose*. We denote by V_N the set of vertices met by a noose N, i.e., $V(N) = V(G) \cap N$. The *length* N of a noose is the number of the vertices that it meets and is denoted by $|N|$, i.e., $|N| = |V(N)|$. The *facewidth* of a Σ-embedded graph embedding (G, τ) is the smallest length of a non-contractible (i.e., non null-homotopic) noose in Σ and is denoted by $\mathbf{fw}(G)$. We call a Σ-embedded graph G *polyhedral* [25] if G is 3-connected and $\mathbf{fw}(G) \geq 3$, or if G is a clique and $1 \leq |V(G)| \leq 3$.

Definition 1. *Let $\alpha, \beta, \gamma, \delta$ be non-negative integers. A graph G is $(\alpha, \beta, \gamma, \delta)$*-nearly embeddable*, if there is a surface Σ of Euler genus γ and a set of vertices $A \subseteq V(G)$ (called* apices*) of size at most α such that graph $G \setminus A$ is the union of subgraphs $\mathcal{G} = \{G_0, \ldots, G_\delta\}$ (some of them may be the empty graph) with the following properties:*

1. *There is a set $\mathcal{R} = \{\Delta_1, \ldots, \Delta_\delta\}$ of pairwise disjoint open discs in Σ such that G_0 is a graph embedded in Σ in a way that $G_0 \cap \bigcup_{i=1,\ldots,\delta} \Delta_i = \emptyset$ (G_0 is called* underlying graph *of G),*
2. *the graphs $G_1, \ldots, G_\delta$ (called* vortices*) are pairwise disjoint and for $1 \leq i \leq \delta$, $V(G_0) \cap V(G_i) \subseteq \mathbf{bor}(\Delta_i)$ (we call the vertices in $V(G_0) \cap V(G_i) \subseteq \mathbf{bor}(\Delta_i)$* base vertices *of the vortex G_i),*
3. *for $1 \leq i \leq \delta$, let $U_i = \{u^i_1, \ldots, u^i_{m_i}\}$ be the base vertices of G_i appearing in an order obtained by counterclockwise traversing $\mathbf{bor}(\Delta_i)$. Then G_i has a path decomposition $\mathcal{B}_i = (B^i_j)_{1 \leq j \leq m_i}$, of width equal to β such that for $1 \leq j \leq m_i$, we have $u^i_j \in B^i_j$. We also denote $\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_\delta\}$ and for each v^i_j, we call B^i_j the* cloud *of v^i_j.*

If G is an $(\alpha, \beta, \gamma, \delta)$-nearly embeddable graph for some $\mathfrak{E} = (A, \Sigma, \mathcal{G}, \mathcal{R}, \mathcal{B})$ as above, we say that $\mathfrak{E}$ is its *embedding pattern*. If in Definition 1 we demand the embedding of the graph G_0 to be polyhedral, then we say that G is *polyhedrally* $(\alpha, \beta, \gamma, \delta)$*-nearly embeddable*, and we say that the corresponding pattern is *polyhedral*. We also say that G is (*polyhedrally*) *c-nearly embeddable* graph if it is (polyhedrally) $(\alpha, \beta, \gamma, \delta)$-nearly embeddable for some $\alpha, \beta, \gamma, \delta \leq c$.

We would like to stress that the proof of Proposition 1 below, which is the main result of this section, heavily relies on the recent result of Kawarabayashi and Wollan [21] to find in time $O(n^3)$ the tree-like decomposition of an H-minor-free graph G, given by structure theorem of Robertson and Seymour [30].

Proposition 1. *There exists an algorithm that, given an h-vertex graph H and an n-vertex graph G that excludes H as a minor, outputs, in $O_h(n^3)$ steps, a tree decomposition $\mathcal{D} = (\mathcal{X} = \{X^t \mid t \in V(T)\}, T)$ of G of adhesion $O_h(1)$ and such that every $t \in V(T)$, $\overline{X}^t$ is a polyhedrally $O_h(1)$-nearly embeddable graph. Moreover, the same algorithm outputs the corresponding embedding pattern $\mathfrak{E}^t$ of $\overline{X}^t$ for each $t \in V(T)$.*

Sketch of proof: The result has been recently been proven in [22] without the demand that $\overline{X}^t$ is a polyhedrally $O_h(1)$-nearly embeddable graph. In fact, the non-polyhedral version of Theorem 1 is known as the Structure Theorem of the Graph Minors series [31]. In [22], a simpler proof of this theorem was found together with an algorithm with the claimed running time (see also [10]). In order to impose the polyhedral condition, we further build a tree decomposition of each $\overline{X}^t$ where all of its bags are polyhedrally $O_h(1)$-nearly embeddable. For this, assume that $\mathfrak{E}^t = (A^t, \Sigma^t, \mathcal{G}^t, \mathcal{R}^t, \mathcal{B}^t)$ is the embedding pattern of a bag $\overline{X}^t$ that is non-polyhedrally $(\alpha, \beta, \gamma, \delta)$-nearly embeddable where $\alpha, \beta, \gamma, \delta \leq c_h$.

We fist update $\mathcal{B}^t$ so that for each $i \in \{1, \ldots, \delta\}$ and $j \in \{1, \ldots, m_i\}$, the following hold: (a) $u^i_j, u^i_{j+1} \in B^i_j$ (here we interpret $u^i_{m_i+1}$ as u^i_1), (b) $u^i_j \in B^i_j \cap B^i_{j-1}$ (here we interpret B^i_0 as $B^i_{m_i}$), and (c) $|B^i_j \cap B^i_{j+1}| \leq \frac{k+1}{2}$ (here we interpret $B^i_{m_i+1}$ as B^i_1).

This can be done if for every $i \in \{1, \ldots, \delta\}$, we set $\mathcal{B}_i \leftarrow (\hat{B}^i_j)_{1 \le j \le m_i}$ so that $\hat{B}^i_j = B^i_j \cup B^i_{j+1}, j \in \{1, \ldots, m_i\}$ (here $B^i_{m_i+1} = B^i_1$). We call the conditions (a), (b), (c) *flexibility conditions for* $\mathcal{B}^t$ and we show that they can be assumed with the cost of updating β to be $2\beta + 1$. This enhancement of $\mathcal{B}^t$ is necessary in order to build a tree decomposition of $\overline{X}^t$. For this, we essentially imitate the steps of Algorithm 1 in [33] by including in the common intersections of the bags of this tree decomposition not only the vertices met by the separating nooses, but also their cloud vertices along with the (common) apex set A. In what follows, we outline this enhancement process.

The fact that $\mathfrak{E}^t$ is not polyhedral implies that the underlying graph G^t_0, embedded in Σ has a noose N where $|V(N)| \le 2$ and such that one of the following holds:

(1) $V(N)$ is a separator of G^t_0.
(2) $V(N)$ is not a separator of G^t_0 and $\mathcal{N}$ is non-contracible and not surface separating.

In case (1), $\overline{X}^t$ has a separation (X, Y) where $X \cap Y$ contains the vertices of $V(N)$ along with their clouds and the apices in A. Making use of the flexibility conditions, it is possible to split the bag $\overline{X}^t$ into two bags A and B that are both $(\alpha, \beta, \gamma, \delta)$-nearly embeddable and the flexibility conditions are still satisfied for both of them. While step (1) is applied, the apices in A^t are included in the common intersection of any two neighboring bags of this tree decomposition of $\overline{X}^t$. This makes the adhesion of the resulting tree decomposition to be at most $2\beta + \alpha$.

In case (2), we revise $\mathfrak{E}^t = (A^t, \Sigma^t, \mathcal{G}^t, \mathcal{R}^t, \mathcal{B}^t)$ to a new one $\mathfrak{E}' = (A^{t\prime}, \Sigma^{t\prime}, \mathcal{G}^{t\prime}, \mathcal{R}^{t\prime}, \mathcal{B}^{t\prime})$ as follows: we define $A^{t\prime}$ so that it contains all vertices in A^t along with the vertices in $V(N)$ and their clouds. We define $\Sigma^{t\prime}$ to be the surface where $G^t_0 \setminus V(N)$ is embedded, and take into account that $\gamma' = \gamma(\Sigma^{t\prime}) < \gamma$. Our next step is to see that N may split, in the new embedding, at most one of the vortices in $\mathcal{G}^t$ to two vortices, and this induces a revision of $(\mathcal{G}^t, \mathcal{R}^t, \mathcal{B}^)$ to a new triple $(\mathcal{G}^{t\prime}, \mathcal{R}^{t\prime}, \mathcal{B}^{t\prime})$ such that $\overline{X}^t$ is $(\alpha', \beta', \gamma', \delta')$-nearly embeddable with pattern $(A^{t\prime}, \Sigma^{t\prime}, \mathcal{G}^{t\prime}, \mathcal{R}^{t\prime}, \mathcal{B}^{t\prime})$ where $\alpha' \le \alpha + 2 \cdot \beta$, $\beta' = \beta$, $\gamma' \le \gamma - 1$, and $\delta' \le \delta + 1$. Also, this update can be done in a way flexibility conditions again hold for $\mathcal{B}^{t\prime}$. Our progress is that we reduced γ' in the cost of increasing α' and δ'. This means that case (2) will not be applied more than $O_h(1)$ times, and therefore after the application of the transformation of cases (1) and (2) until this is not possible anymore, we end up with a tree decomposition of adhesion $2\beta + \alpha = O_h(1)$ consisting of polyhedrally $(\alpha', \beta', \gamma', \delta')$-nearly embeddable bags, where $\alpha', \beta, \gamma', \delta' = O_h(1)$. Given that we have the decomposition $\mathcal{D}$ and the embedding pattern $\mathfrak{E}^t$ of $\overline{X}^t$ for each $t \in V(T)$ and following the arguments of the proof of Proposition 4.2 in [33], it is easy to verify that, given $\mathcal{D}$ and the embedding pattern $\mathfrak{E}^t$ of $\overline{X}^t$ for each $t \in V(T)$, the procedure described in the above sketch requires at most $O(n^3)$ steps. □

Given an h-vertex graph H and an H-minor free graph G, we call a tree decomposition $\mathcal{D}$ as the one in Proposition 1, a *c_h-nearly polyhedral decomposition* (where c_h is a constant depending only on h). If in Proposition 1 G is a graph embedded in a surface, then a *c_h-nearly polyhedral decomposition* is what has been defined in [33] as a *polyhedral decomposition*, where the adhesion is at most 2 (now in each embedding pattern $\mathfrak{E}^t = (A^t, \Sigma^t, \mathcal{G}^t, \mathcal{R}^t, \mathcal{B}^t)$, we have that $\mathcal{G}^t_0$ contains only the underlying graph, while $\mathcal{R}^t = \emptyset$ and $\mathcal{B}^t = \emptyset$).

5 H-minor-free cut decompositions

In this section we define and show how to construct a special type of branch decompositions for families of graphs excluding a graph H as a minor; we call them *H-minor-free* *cut decompositions*. Their construction relies on surface cut decompositions, recently introduced in [33].

Let Σ be a surface without boundary, and let $\mathcal{N}$ be a finite set of O-arcs in Σ pairwise intersecting at zero-dimensional subsets of Σ (i.e., points). Then the closure of each connected component of $\Sigma \setminus \mathbf{U}\mathcal{N}$ is called a *pseudo-surface*, where $\mathbf{U}\mathcal{N} = \bigcup_{N\in\mathcal{N}} N$. For a point $p \in \Sigma$, let $\mathcal{N}(p)$ be the number of O-arcs in $\mathcal{N}$ containing p, and let $P(\mathcal{N}) = \{p \in \Sigma : \mathcal{N}(p) \geq 2\}$; note that by assumption $P(\mathcal{N})$ is a finite set of points of Σ. Then we define $\theta(\mathcal{N}) = \sum_{p\in P(\mathcal{N})}(\mathcal{N}(p) - 1)$.

Note that if the O-arcs are pairwise disjoint, then each pseudo-surface is a surface with boundary. Notice that in Definition 1 we can permit Σ to be a pseudo-surface with boundary instead of a surface without boundary. This extension of the definition is necessary for defining the concept of an H-minor-free cut decomposition below.

Definition 2. *Given an h-vertex graph H, an n-vertex H-minor-free graph G, an* H-minor-free cut decomposition *of G is a branch decomposition (T, μ) of G such that there exists an $O_h(1)$-nearly polyhedral decomposition $\mathcal{D} = (\mathcal{X} = \{X^t \mid t \in V(T')\}, T')$ of G such that for each edge $e \in E(T)$, either $|\mathbf{mid}(e)| = O_h(1)$ or there exists a bag $X^t \in \mathcal{X}$ such that*

- $\mathbf{mid}(e) \subseteq V(X^t)$*;*
- *given that $\mathfrak{E}^t = (A^t, \Sigma^t, \mathcal{G}^t, \mathcal{R}^t, \mathcal{B}^t)$ is the embedding pattern of $\overline{X}^t$ and $\mathcal{G}^t = \{G_0^t, G_1^t, \ldots, G_\delta^t\}$, there exists a set $\mathcal{N}$ of nooses of G_0^t in Σ^t such the vertices of $\mathbf{mid}(e) \cap V(G_0^t)$ are all met by the nooses in $\mathcal{N}$ in a way that the following hold*
  1. $|\mathcal{N}| = O_h(1)$*,*
  2. *the nooses in $\mathcal{N}$ pairwise intersect only at subsets of $\mathbf{mid}(e)$,*
  3. $\theta(\mathcal{N}) = O_h(1)$*,*
  4. *$\Sigma^t \setminus \mathbf{U}\mathcal{N}$ contains exactly two connected components, such that the graph $\overline{G}[V(G_e) \cap V(X^t)]$ is $O_h(1)$-nearly embedded in the closure of one of them.*

If in the above definition we consider that G is embedded in some surface of genus γ and we restrict each $\mathcal{G}^t$ to contain only the underlying graph (i.e., there are no vortices, $\mathcal{R}^t = \emptyset$, and $\mathcal{B}^t = \emptyset$), we have the definition of surface cut decompositions introduced in [33]. Finally, if we further restrict Σ to be a sphere and set $A^t = \emptyset$, we have the notion of sphere cut decompositions introduced in [14, 34].

Theorem 3. *There exists an algorithm that, given an h-vertex graph H and an n-vertex graph G that excludes H as a minor and has branchwidth at most k, outputs in $O_h(n^3)$ steps an H-minor-free cut decomposition of G of width $O_h(k)$.*

Sketch of proof: The algorithm applies first the algorithm of Proposition 1 in order to find a tree decomposition $\mathcal{D} = (\mathcal{X} = \{X^t \mid t \in V(T')\}, T')$ of adhesion $O_h(1)$ and such that for every $t \in V(T')$, $\overline{X}^t$ is a polyhedrally $O_h(1)$-nearly embeddable graph. Then, for every $t \in V(T')$ we consider the underlying graph G_0^t, and find a branch decomposition of it of width $O_h(k)$. This can be done in $O(n^3)$ steps by using a standard planarization procedure that cuts the graph along minimum length non-contractible nooses (see [11, 13]) and then using the approximation algorithm in [34]. Our next aim is to transform this branch decomposition into a new branch decomposition (T^t, τ^t) of G_0^t with the property that for every $e \in E(T)$, there exists a set $\mathcal{N}$ of nooses of G_0^t in Σ^t such that the vertices of $\mathbf{mid}(e) \cap V(G_0^t)$ are all met by the nooses in $\mathcal{N}$, in a way that conditions *1–3* of Definition 2 hold and, moreover, $\Sigma^t \setminus \mathbf{U}\mathcal{N}$ contains exactly two connected components, such that the graph $G_e \setminus A$ is embedded in the closure of one of them. This construction is using Algorithm 2 of [33] and the proof of its correctness uses the same arguments as the proof of Theorem 7.2 in [33]. Our next step is to enhance (T^t, τ^t) so that it becomes a branch decomposition of $\overline{G}[V(G_e) \cap V(X^t)]$ satisfying property *4*. For this, we use Observation 5.1 of [33] to enhance (T^t, τ^t) so to contain also all edges incident with the vertices in A^t, and then we use the construction of Lemma 10 of [13] in order to further enhance it to a branch decomposition $(\hat{T}^t, \hat{\tau}^t)$ of $\overline{G}[V(G_e) \cap V(X^t)]$ so that property *4* holds. Our last step is to glue together all branch decompositions $(T^t, \tau^t), t \in T'$ by following the way the bags of $\mathcal{X}$ are glued together in $\mathcal{D}$. The outcome of this procedure gives the required H-minor-free cut decomposition of G. □

In contract with the algorithm of Theorem 3, it is worth noting here that the algorithm that computes a surface cut decomposition for a surface-embedded graph in [33] has running time $2^{O(k)} \cdot n^3$, because we wanted to optimize the dependence on the genus of the width of the obtained branch decomposition, while keeping the overall running time single-exponential in k.

6 Combinatorial behavior of the vortices

In this section we focus on the combinatorial behavior of the vortices in a graph excluding a graph H as a minor. The main objective is to prove that, in an H-minor-free cut decomposition, the number of ways that connected subgraphs can behave inside a vortex can be upper-bounded by the number of $O_h(1)$-packings (defined in Section 2) of size linear in the branchwidth of the input graph. By Theorem 3, from now on we assume that we have an H-minor-free cut decomposition (T, μ) of G, as well as a tree decomposition $\mathcal{D} = (\mathcal{X} = \{X^t \mid t \in V(T)\}, T)$ of G of adhesion $O_h(1)$, such that each $\mathcal{D}$-closure $\overline{X}^t$ is a polyhedrally $(\alpha, \beta, \gamma, \delta)$-nearly embedded graph, with $\alpha, \beta, \gamma, \delta = O_h(1)$.

In order to have a clearer picture of the behavior of the vortices, we define according to [13] the graph $R_{d,s}$, where

$$
\begin{aligned}
V(R_{d,s}) &= V_1 \cup \cdots \cup V_s \text{ with } |V_i| = d \text{ for } 1 \le i \le s, \text{ and} \\
E(R_{d,s}) &= \{\{x_j, x_k\} \mid x_j, x_k \in V_i, 1 \le j \ne k \le d, 1 \le i \le s\} \cup \\
&\quad \{\{x_j, y_j\} \mid x_j \in V_{i-1} \text{ and } y_j \in V_i, 1 \le j \le d, 1 \le i \le s\}.
\end{aligned}
$$

We call such a graph $R_{d,s}$ a *normalized vortex.* In the graph $R_{d,s}$ we distinguish a subset $U \subseteq V(R_{d,s})$ containing exactly one vertex from each V_i. The pair $(R_{d,s}, U)$ is called an *(d,s)-vortex pattern.* See Fig. 2 for an example of a normalized vortex $R_{5,12}$ and the corresponding $(5, 12)$-vortex pattern, where the set U is defined by the white vertices. These white vertices are the *base vertices* of the vortex, and each one of the d concentric paths is called a *track* of the vortex. We say that an (d,s)-vortex pattern has *depth* d. Note that each base vertex belongs to a clique of size d. The d edges between two consecutive cliques are called a *section* of the vortex. Normalized vortices and vortex patterns are useful because any vortex is a minor of a vortex pattern, as stated in the following lemma.

Lemma 2 (Dorn, Fomin, and Thilikos [13]). *Any vortex of a d-nearly embeddable graph with base set J is a minor of a (d,s)-vortex pattern $(R_{d,s}, U)$, where the minor operations map bijectively the vertices of U to the vertices in J in a way that the order of the vortex and the cyclic ordering of U induced by the indices of its elements is respected.*

By Lemma 2, from now on we will only deal with vortex patterns. We say that connected subgraph B of G *meets* a vortex F, if B contains some of the base vertices of F. If U is the set of base vertices of F, the number of times that B meets F is exactly $|V(B) \cap U|$. For the analysis, we need to consider the possibility that a subgraph in

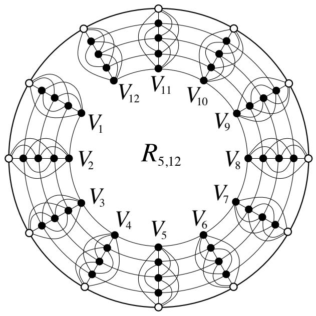

Fig. 2. A $(5, 12)$-vortex pattern. The set U is defined by the white vertices.

a connected packing $\mathcal{P} \in \Psi_G(S)$ meets more than one vortex. This possibility is ruled out in Lemma 3. Loosely speaking, the proof of Lemma 3 is based on the fact that if a subgraph B in a connected packing $\mathcal{P} \in \Psi_G(S)$ meets two vortices of depth at most β, these two vortices can be virtually merged along a path of B into a new vortex of depth at most 2β. As a subgraph may a priori meet an arbitrary number of the vortices, for our analysis we need to consider all possibilities of merging any subset of the δ vortices, which are at most $p(\delta)$ many (see "partitions of an integer" in Section 2). Therefore, potentially some of this merged vortices may have depth up to $\delta \cdot \beta = O_h(1)$. Also, in order to find an upper bound for the number of connected packings, we will need to incur an additional factor $p(\delta)$. Lemma 4, whose proof uses Lemma 3, will allow us to simulate the behavior of the vortices with simpler objects of appropriate size, independent of the integer s (recall that we deal with (β, s)-vortex patterns, cf. Section 6). More details follow.

Lemma 3. *We can assume that each subgraph in a connected packing $\mathcal{P} \in \Psi_G(S)$ meets at most one vortex.*

Sketch of proof: Let B be a subgraph in a connected packing $\mathcal{P} \in \Psi_G(S)$, and assume that B intersects two vortices F_1 and F_2 of depth at most β. Let Δ_1 and Δ_2 be the open discs of Σ such that F_i is contained in Δ_i for $i \in \{1, 2\}$. Since B intersects both F_1 and F_2, we can assume that B contains a path P in Σ connecting the discs Δ_1 and Δ_2, disjoint from all the other vortices. Since G_0 is embedded in Σ, no other subgraph in the connected packing $\mathcal{P}$ intersects the path P. Therefore, the same packing $\mathcal{P}$ would also exist if the two discs Δ_1 and Δ_2 are merged along the path P. In terms of the vortices, this merging operation can be translated into considering a new vortex $F_{1,2}$ of depth at most 2β made of the union of F_1 and F_2 along P, as illustrated in Fig. 3. We omit the details here (see [13] for a similar trick). As a subgraph may a priori intersect an arbitrary number of vortices, we need to consider all possibilities of merging any subset of the δ vortices, which are at most $p(\delta)$ many (see Section 2). Therefore, potentially some of this merged vortices may have depth up to $\delta \cdot \beta = O_h(1)$. □

Lemma 4. *For each vortex F, we can assume that the total number of times that the subgraphs in a connected packing $\mathcal{P} \in \Psi_G(S)$ meet F is $O_h(k)$.*

Sketch of proof: Again, it will be useful to assume that the vortices are all isomorphic to (β, s)-vortex patterns (see Fig. 2). Let F be a fixed vortex, and let us order its base vertices counterclockwise as $U = \{u_1, u_2, \ldots, u_s\}$ (cf. the white set of vertices in Fig. 2).

We can also assume that each subgraph in a connected packing is a tree of G, which is enough for connectivity purposes. Each leaf of such a tree is a vertex in the corresponding

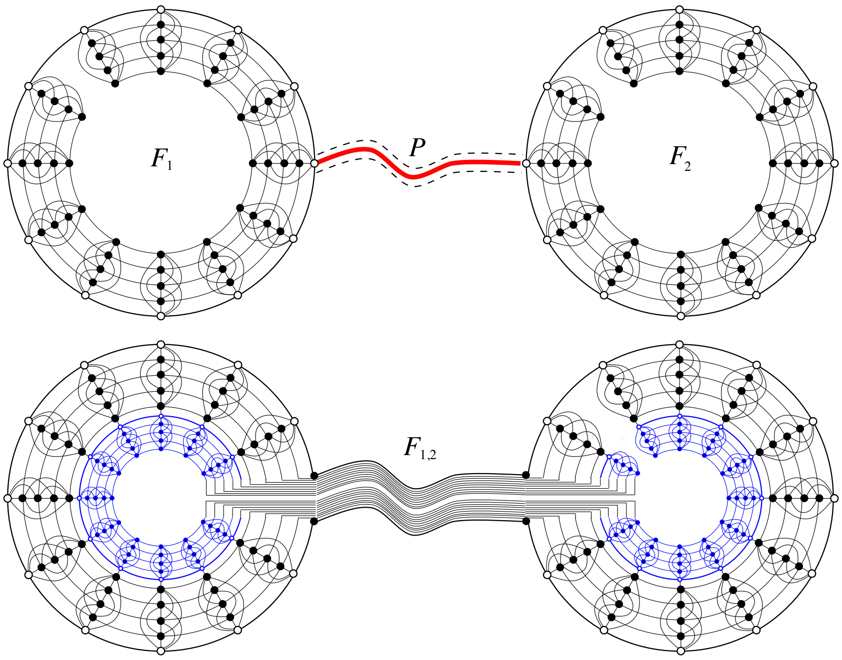

Fig. 3. Example in the proof of Lemma 3. If a subgraph intersects two vortices F_1 and F_2 (illustrated above with the red path P), we can consider a bigger vortex $F_{1,2}$ (depicted below) obtained from merging vortices F_1 and F_2 along the path P.

middle set of the branch decomposition. As the total number of vertices in each middle set is at most k, in order to prove that a given vortex is met by some subgraph $O_h(k)$ times, it is enough to prove that every path of a subgraph with both endvertices being in the separator S meets a given vortex $O_h(1)$ times. Therefore, we assume henceforth that the subgraphs in a connected packing are paths. Throughout the proof, see Fig. 4. In the figure, vertices of the form v_i belong to the separator S of the minor-free cut decomposition, while vertices of the form u_j are base vertices of the vortex.

Let B a fixed path intersecting F (cf. the thick black path in Fig. 4, containing vertices v_1 and v_4), and let u_f (resp. u_l) be the first (resp. last) vertex of $V(B) \cap U$ according to the counterclockwise ordering of U (in Fig. 4, we have $u_f = u_2$ and $u_l = u_{12}$). By Lemma 3, we may assume that B does not intersect any other vortex other than F. Let u_i and u_j be two consecutive vertices in $V(B) \cap U$, with $f < i < j < l$. If no other subgraph intersects a vertex $u_\ell \in U$, with $i < k < j$ (cf. vertices u_4 and u_6 in Fig. 4), we can just shortcut the path B from the clique containing u_i to the clique containing u_j along a free track of the vortex, so that B does not intersect u_i or u_j anymore (such a shortcut is depicted with a thick green path in Fig. 4). This transformation preserves the connectivity of B and does not modify any other subgraph in the connected packing. Therefore, we may assume that some other subgraph B' intersects a vertex $u_\ell \in U$, with $i < k < j$. But since B is connected, B contains a path P from u_i or u_j outside the vortex H (cf. vertices u_8 and u_{11} in Fig. 4). The idea is that if B' intersects u_ℓ and is disjoint from P (by definition of a connected packing), then necessarily B' uses a handle or a noose (for instance, in Fig. 4, the red path containing vertex v_2 uses a handle, and the blue path containing vertex v_3 uses a noose). There may be other vortices F' inside the region delimited by P (cf. vortex F' in Fig. 4), but by Lemma 3 we can assume that no subgraph intersecting F' intersects also F. That is, the number of times that the path B intersects the vortex F can be linearly upper-bounded by the total number of

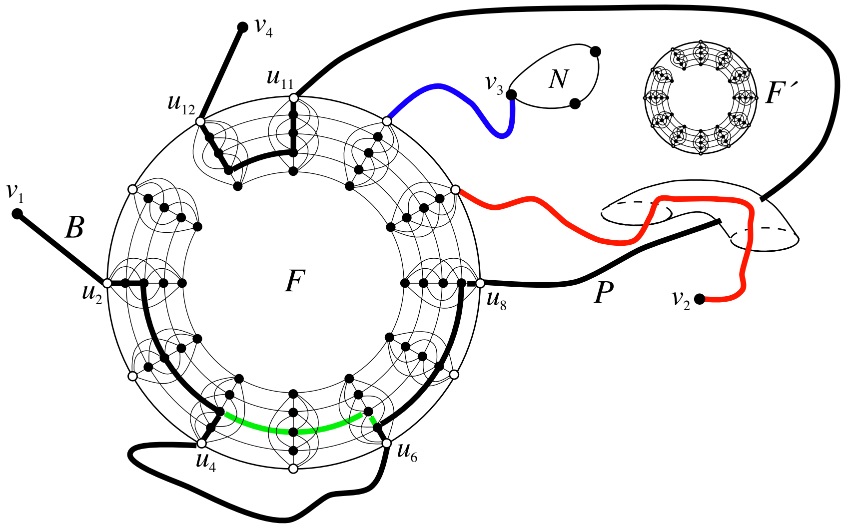

Fig. 4. Example in the proof of Lemma 4.

handles and nooses in the surface. Since the genus of Σ is $O_h(1)$, as well as the number of nooses in an H-minor-free cut decomposition, we conclude that B meets F at most $O_h(1)$ times. The lemma follows. □

Let F be a given (d, s)-vertex pattern with set of base vertices $U = \{u_1, \ldots, u_s\}$, ordered cyclically. A *configuration* in F is a set of vertex-disjoint subgraphs $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ of F. We say that two subgraphs $F_i, F_j \in \mathcal{F}$ *cross* if there exist $u_{i_1}, u_{i_2} \in V(F_i) \cap U$ and $u_{j_1}, u_{j_2} \in V(F_j) \cap U$ such that $i_1 < j_1 < i_2 < j_2$. The *crossing graph* $\mathcal{F}_c$ of a configuration $\mathcal{F}$ has one vertex for each subgraph in $\mathcal{F}$, and an edge between two vertices if and only if their corresponding subgraphs cross. A configuration $\mathcal{F}$ is said to be an *ℓ-configuration* if the maximum size of a clique in $\mathcal{F}_c$ is ℓ. In the following lemma we prove that in a vertex of given depth, the existing configurations can cross only in a *bounded* way. This fact will enable us to upper-bound the number of configurations in a vortex of depth d in terms of the number of d-packings in the circle.

Lemma 5. *A vortex pattern of depth at most β does not contain any β'-configuration with $\beta' > \beta$.*

Sketch of proof: Let F be a vortex pattern of depth d, and assume that $\mathcal{F}$ is a configuration in F. The key observation is that is F_1 and F_2 are two crossing subgraphs in $\mathcal{F}$, then by the structure of a vortex pattern necessarily F_1 and F_2 use two edges in a same section of F. Inductively, one can easily check that if $F_1, \ldots, F_\ell$ are pairwise crossing subgraphs in $\mathcal{F}$, then necessarily $F_1, \ldots, F_\ell$ use ℓ edges in a same section of F. Therefore, as the number of edges in a section of F is exactly d, it follows that $\mathcal{F}_c$ does not contain any clique of size at least $d + 1$. □

7 Upper-bounding the size of the tables

In this section we show that by using H-minor-free cut decompositions in order to solve connected packing-encodable problems in H-minor-free graphs, one can guarantee single-exponential upper bounds on the size of the tables of dynamic programming algorithms.

Theorem 1 follows directly by the definition of a connected packing-encodable problem and the following lemma, which we will prove in this section.

Lemma 6. *Let G be a graph excluding an h-vertex graph H as a minor, and let (T,μ) be an H-minor-free cut decomposition of G of width at most k. Then for every $e \in E(T)$, $|\Psi_{G_e}(\mathbf{mid}(e))| = 2^{O_h(k)}$.*

Let (T,μ) be an H-minor-free cut decomposition of a graph G. For edges $e \in E(T)$ such that $\mathbf{mid}(e) = O_h(1)$, we trivially have that $|\Psi_{G_e}(\mathbf{mid}(e))| = 2^{O_h(1)}$, and therefore the statement of Lemma 6 is satisfied. Therefore, we only need to deal with edges $e \in E(T)$ such that $\mathbf{mid}(e)$ is contained in a graph which is polyhedrally $O_h(1)$-nearly embedded in a surface Σ.

In order to prove Lemma 6, we will need the lemmata of Section 6 about the combinatorial behavior of the vortices. We will also need the following key result from [33], which bounds the size of the tables for graphs embedded in surfaces with boundary.

Lemma 7 (Rué, Sau, and Thilikos [33]). *Let G be a graph containing a set A of vertices such that $G \setminus A$ is embedded in a surface Σ with boundary. Let also S be the set of vertices of G that lie on the boundary of Σ and $A' \subseteq A$. Then, if $|S| \leq k$ and $|A|, \gamma(\Sigma), \nu(\Sigma) \leq \gamma$, then $|\Psi_G(S \cup A')| = \gamma^{O(\gamma)} \cdot k^{O(\gamma)} \cdot \gamma^{O(k)}$.*

Note that, in the statement of Lemma 7, if $|A|, \gamma(\Sigma), \nu(\Sigma) = O_h(1)$, then it holds that $|\Psi_G(S \cup A')| = 2^{O_h(k)}$. The following lemma gives an upper bound on the number of non-crossing packings on a surface with apices and vortices. Intuitively, our approach consists in considering each vortex as a new virtual noose of length $O_h(k)$ in an H-minor-free cut decomposition, where each vertex of such noose corresponds to an eventual meeting of a subgraph of the connected packing with the corresponding vortex. We then consider all non-crossing packings taking into account the original and the virtual nooses, and finally we merge the subgraphs incident to a virtual noose according to the possible $O_h(1)$-packings corresponding to that vortex. This approach is made more precise in Lemma 8 below, which implies Lemma 6, and therefore also Theorem 1 and Theorem 2. The proof of Lemma 8 makes use of Lemmata 2, 4, 5, and 7.

Lemma 8. *Let G be a graph polyhedrally $(\alpha,\beta,\gamma,\delta)$-nearly embedded in a surface Σ with boundary, with a set of apices A. Let also S be the set of vertices of G that lie on the boundary of Σ. If $|S| \leq k$ and $\alpha, \beta, \gamma, \delta, \nu(\Sigma) \leq \eta$, then $|\Psi_G(S \cup A)| = 2^{O_\eta(k)}$.*

Sketch of proof: Let $\mathcal{G} = \{G_0, G_1, \ldots, G_\delta\}$ be the subgraphs of $G \setminus A$ defining its polyhedral $(\alpha,\beta,\gamma,\delta)$-nearly embedding. By Lemma 2, we can assume that each vortex is isomorphic to an (β, s)-vortex pattern for some $s \geq 1$. Recall that each vortex is contained in an open disc Δ_i of Σ, such that $U_i = V(G_0) \cap V(G_i) \subseteq \mathbf{bor}(\Delta_i)$. Also, by Lemma 4, we can assume that the subgraphs of a connected packing meet each vortex $O_h(k)$ times. Let $S = \{v_1, \ldots, v_k\}$ be the set of vertices of G that lie on the boundary of Σ, which we can assume to be contained in a set of disjoint nooses $N_1, \ldots, N_\gamma$ with $\gamma = O_h(1)$ (see [33] for more details). Let c_h be a constant such that $c_h k$ satisfies the statement of Lemma 4. For each vortex G_i, $1 \leq i \leq \delta$, we consider a set of $c_h k$ auxiliary vertices $S_i = \{u_i^1, \ldots, u_i^{c_h k}\}$ placed cyclically inside around Δ_i, corresponding to the potential meetings of the subgraphs in a connected packing with the vortex. We see each set of auxiliary vertices S_i as a new noose $\bar{N}_i$ of size $c_h k$, placed in Σ inside the border of Δ_i, such that both nooses are concentric. Potentially, the vertices in U_i (which are in $\mathbf{bor}(\Delta_i)$) can be joined to the vertices in S_i (which are in $\bar{N}_i$) in any *planar* way inside Σ. In order to formalize this fact, we define the class of graphs $\bar{\mathcal{G}}_0$ as follows: $\bar{\mathcal{G}}_0$ contains all the graphs that can be obtained from G_0 by replacing each vortex G_i with

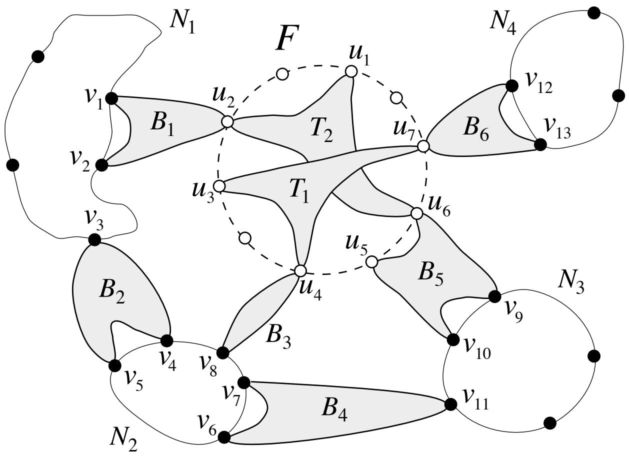

Fig. 5. Example in the plane of our approach to simulate the behavior of the vortices. There are four nooses N_1, N_2, N_3, N_4 drawn with full lines, and a vortex F of depth 2 drawn with a dashed circle. Black vertices correspond to vertices in the separator S (thus, in the nooses), while white vertices belong to the base set of the vortex. The non-crossing packing $\mathcal{P}_0$ in Σ has the following six subgraphs (depicted with dark regions), abstracted as blocks containing only black and white vertices: $B_1 = \{v_1, v_2, u_2\}, B_2 = \{v_3, v_4, v_5\}, B_3 = \{v_8, u_4\}, B_4 = \{v_6, v_7, v_{11}\}, B_5 = \{v_9, v_{10}, u_5, u_6\}$, and $B_6 = \{v_{12}, v_{13}, u_7\}$. With the two subgraphs $T_1 = \{u_3, u_4, u_7\}$ and $T_2 = \{u_1, u_2, u_6\}$ corresponding to a 2-packing of the vortex, subgraphs B_1 and B_5 (resp. B_6 and B_6) get merged into the new subgraph $\{v_1, v_2, v_9, v_{10}\}$ (resp. $\{v_8, v_{12}, v_{13}\}$).

a planar bipartite subgraph $\bar{G}_i$ with bipartition U_i and S_i, such that the vertices of U_i are placed around $\mathbf{bor}(\Delta_i)$, and the vertices of S_i are placed around $\bar{N}_i$. (We would like to stress that do not need to compute this class of graphs algorithmically, it is only used in the analysis to provide an upper bound on the number of connected packings.) Note that each graph $\bar{G}_0 \in \bar{\mathcal{G}}_0$ is embedded in Σ, and therefore Lemma 7 can be applied to $\bar{G}_0$. Let $\bar{S} = S \cup \left(\bigcup_{i=1}^{\delta} S_i\right)$, and note that $|\bar{S}| \leq k + O_h(k) = O_h(k)$, and that the vertices in $\bar{S}$ are contained in a set of at most $\gamma + \delta = O_h(1)$ disjoint nooses of Σ. The key observation is that every connected packing $\mathcal{P} \in \Psi_G(S \cup A)$ can be obtained by combining a non-crossing packing $\mathcal{P}_0 \in \Psi_{\bar{G}_0}(\bar{S} \cup A)$, for some graph $\bar{G}_0 \in \bar{\mathcal{G}}_0$, with appropriate $O_h(1)$-packings of the (possibly merged) vortices. Indeed, by Lemma 5 we know that a vortex pattern of depth at most β does not contain any β'-configuration with $\beta' > \beta$, so the number of β-packings on $c_h k$ elements gives an upper bound to the number of configurations in each vortex. See Fig. 5 for an example of such construction.

Hence, from the above discussion we conclude that

$$
\begin{aligned}
|\Psi_G(S \cup A)| &\leq \max_{\bar{G}_0 \in \bar{\mathcal{G}}_0} \{|\Psi_{\bar{G}_0}(\bar{S} \cup A)|\} \cdot p(\delta) \cdot \left(P_{\delta\beta}(c_h k)\right)^{\delta} \\
&= 2^{O_h(k)} \cdot 2^{O_h(1)} \cdot \left(2^{O_h(k)}\right)^{O_h(1)} \;=\; 2^{O_h(k)} \;,
\end{aligned}
$$

where in the last equality we have used Lemma 7, the estimate for $p(q)$ given in Section 2, and Lemma 1, in this order. The result follows. □

References

1. G. Andrews. *The Theory of Partitions.* Cambridge University Press, 1984.
2. S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability – a survey. *BIT*, 25(1):2–23, 1985.
3. V. Arvind and P. Mukhopadhyay. Derandomizing the Isolation Lemma and Lower Bounds for Circuit Size. In *Proc. of APPROX-RANDOM*, volume 5171 of *LNCS*, pages 276–289, 2008.
4. 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, 1988.
5. P. Bonsma. Surface split decompositions and subgraph isomorphism in graphs on surfaces. *CoRR*, abs/1109.4554, 2011, to appear in the *Proc. of STACS 2012*.
6. L. Cai and D. W. Juedes. On the existence of subexponential parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):789–807, 2003.
7. 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.
8. M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proc. of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 150–159, 2011.
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. T. Hajiaghayi, and K.-i. Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 637–646, 2005.
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 decompositions. *Algorithmica*, 58(3):790–810, 2010.
15. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
16. A. Dress, J. H. Koolen, and V. Moulton. On line arrangements in the hyperbolic plane. *European Journal of Combinatorics*, 23(5):549–557, 2002.
17. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer Verlag, 2006.
18. M. Grohe and D. Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *CoRR*, abs/1111.1109, 2011.
19. R. Impagliazzo, R. Paturi, and F. Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
20. J. Jonsson. Generalized triangulations and diagonal-free subsets of stack polyominoes. *Journal of Combinatorial Theory, Series A*, 112(1):117–142, 2005.
21. K.-i. Kawarabayashi and P. Wollan. A simpler algorithm and shorter proof for the graph minor decomposition. In *Proc. of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 451–458, 2011.
22. K.-i. Kawarabayashi and P. Wollan. A simpler algorithm and shorter proof for the graph minor decomposition. In *Proc. of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 451–458, 2011.

23. D. Lokshtanov, D. Marx, and S. Saurabh. Known Algorithms on Graphs of Bounded Treewidth are Probably Optimal. In *Proc. of the 22nd annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 777–789, 2011.
24. D. Lokshtanov, D. Marx, and S. Saurabh. Slightly Superexponential Parameterized Problems. In *Proc. of the 22nd annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 760–776, 2011.
25. B. Mohar and C. Thomassen. *Graphs on surfaces*. John Hopkins University Press, 2001.
26. K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.
27. T. Nakamigawa. A generalization of diagonal flips in a convex polygon. *Theoretical Computer Science*, 235(2):271–282, 2000.
28. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
29. N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to Tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.
30. N. Robertson and P. D. Seymour. Graph minors. XVI. Excluding a Non-planar Graph. *Journal of Combinatorial Theory, Series B*, 77:1–27, 1999.
31. N. Robertson and P. D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003.
32. J. Rué, I. Sau, and D. M. Thilikos. Asymptotic enumeration of non-crossing partitions on surfaces. *CoRR*, abs/1104.2477, 2011 (a preliminary version appeared in the *Proc. of ICALP 2011*). Last version available at `www.lirmm.fr/∼sau/Pubs/RST11NCP.pdf`.
33. J. Rué, I. Sau, and D. M. Thilikos. Dynamic programming for graphs on surfaces. *CoRR*, abs/1104.2486, 2011, to appear in *ACM Transactions on Algorithms (TALG)*. Last version available at `www.lirmm.fr/∼sau/Pubs/RST12TALG.pdf`.
34. P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.
35. S. Tazari. Faster approximation schemes and parameterized algorithms on (odd-)H-minor-free graphs. *Theoretical Computer Science*, 417:95–107, 2012.