A Tight Subexponential-time Algorithm for Two-Page Book Embedding

ROBERT GANIAN, Algorithms and Complexity Group, TU Wien, Austria
HAIKO MÜLLER, School of Computing, University of Leeds, United Kingdom
SEBASTIAN ORDYNIAK, School of Computing, University of Leeds, United Kingdom
GIACOMO PAESANI, Dipartimento di Informatica, Sapienza University of Rome, Italy
MATEUSZ RYCHLICKI, School of Computing, University of Leeds, United Kingdom

A book embedding of a graph is a drawing that maps vertices onto a line and edges to simple pairwise non-crossing curves drawn into "pages", which are half-planes bounded by that line. Two-page book embeddings, i.e., book embeddings into 2 pages, are of special importance as they are both NP-hard to compute and have specific applications. We obtain a $2^{O(\sqrt{n})}$ algorithm for computing a book embedding of an n-vertex graph on two pages—a result which is asymptotically tight under the Exponential Time Hypothesis. As a key tool in our approach, we obtain a single-exponential fixed-parameter algorithm for the same problem when parameterized by the treewidth of the input graph. We conclude by establishing the fixed-parameter tractability of computing minimum-page book embeddings when parameterized by the feedback edge number, settling an open question arising from previous work on the problem.

CCS Concepts: • **Theory of computation → Parameterized complexity and exact algorithms**; • **Human-centered computing → Graph drawings**.

Additional Key Words and Phrases: (two-page) book embedding, subhamiltonian cycle, Hamiltonian cycle, treewidth, parameterized complexity, subexponential-time algorithms, graph drawing, planar graph, SPQR-tree, sphere-cut decomposition

ACM Reference Format:
Robert Ganian, Haiko Müller, Sebastian Ordyniak, Giacomo Paesani, and Mateusz Rychlicki. 2025. A Tight Subexponential-time Algorithm for Two-Page Book Embedding. 1, 1 (July 2025), 41 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Book embeddings of graphs are drawings centered around a line, called the *spine*, and half-planes bounded by the spine, called *pages*. In particular, a k-page book embedding of a graph G is a drawing which maps vertices to distinct points on the spine and edges to simple curves on one of the k pages such that no two edges on the same page cross [6]. These embeddings have been the focus of extensive study to date [16, 20–22, 25, 42, 53], among others due to their classical applications in VLSI, bio-informatics, and parallel computing [11, 20, 32].

Every n-vertex graph is known to admit an $\lceil \frac{n}{2} \rceil$-page book embedding [6, 11, 31], but in many cases it is possible to obtain book embeddings with much fewer pages. Particular attention has been paid to two-page embeddings, which have specifically been used, e.g., to represent RNA pseudoknots [32, 47]. The class of graphs that can be embedded on

Authors' Contact Information: Robert Ganian, rganian@gmail.com, Algorithms and Complexity Group, TU Wien, Vienna, Austria; Haiko Müller, h.muller@leeds.ac.uk, School of Computing, University of Leeds, Leeds, United Kingdom; Sebastian Ordyniak, sordyniak@gmail.com, School of Computing, University of Leeds, Leeds, United Kingdom; Giacomo Paesani, giacomopaesani@gmail.com, Dipartimento di Informatica, Sapienza University of Rome, Rome, Italy; Mateusz Rychlicki, mkrychlicki@gmail.com, School of Computing, University of Leeds, Leeds, United Kingdom.

Manuscript submitted to ACM

two pages was studied by Di Giacomo and Liotta [27], Heath [33] as well as by other authors [1], and was shown to be a superclass of planar graphs with maximum degree at most 4 [5].

While two-page book embeddings are a special class of planar embeddings, they are not polynomial-time computable unless P = NP. Indeed, a graph admits a two-page book embedding if and only if it is *subhamiltonian* (*i.e.*, is a subgraph of a planar Hamiltonian graph) [6] and testing subhamiltonicity is an NP-hard problem [11]. On the other hand, the aforementioned problem of constructing a two-page book embedding (or determining that none exists)—which we hereinafter call Two-Page Book Embedding—becomes linear-time solvable if one is provided with a specific ordering of the n vertices of the input graph along the spine [32]. While Two-Page Book Embedding can be seen to admit a trivial brute-force $2^{O(n \cdot \log n)}$ algorithm, it has also been shown to be solvable in $2^{O(n)}$ time—in particular, one can branch to determine the allocation of edges into the two pages and then solve the problem via dynamic programming on SPQR trees [2, 35, 36].

Contribution. As our main contribution, we break the single-exponential barrier for Two-Page Book Embedding by providing an algorithm that solves the problem in $2^{O(\sqrt{n})}$ time. Our algorithm is exact and deterministic, and avoids the single-exponential overhead of branching over edge allocations to pages by instead attacking the equivalent subhamiltonicity testing formulation of the problem. It is also asymptotically optimal under the Exponential Time Hypothesis [38]: there is a well-known quadratic reduction that excludes any $2^{o(\sqrt{n})}$ algorithm for Hamiltonian Cycle on cubic planar graphs [26], and a linear reduction from that problem (under the same restrictions) to subhamiltonicity testing [52] then excludes any $2^{o(\sqrt{n})}$ algorithm for our problem of interest.

The central component of our result is a non-trivial dynamic programming procedure that solves Two-Page Book Embedding in time $2^{O(tw)} \cdot n$, where tw is the treewidth of the input graph. The desired subexponential algorithm then follows by the well-known fact that n-vertex planar graphs have treewidth at most $O(\sqrt{n})$ [28, 43, 49]. But in addition to that, we believe our single-exponential treewidth-based algorithm to be of independent interest also in the context of parameterized algorithmics [13, 19].

Indeed, while Two-Page Book Embedding was already shown to be fixed-parameter tractable w.r.t. treewidth (i.e., to admit an algorithm running in time $f(tw) \cdot n$) by Bannister and Eppstein [3], that result crucially relied on Courcelle's Theorem [12]. More specifically, they showed that the required property can be encoded via a constant-size sentence in Monadic Second Order logic, which suffices for fixed-parameter tractability—but unfortunately not for a single-exponential algorithm, and a direct dynamic programming algorithm based on the characterization employed there seems to necessitate a parameter dependency that is more than single-exponential. Moreover, it is not at all obvious how one could employ convolution-based tools—which have successfully led to $2^{O(tw)} \cdot n$ algorithms for, e.g., Hamiltonian Cycle [10, 14, 15]—for our problem of interest here.

Instead, we obtain our results by employing dynamic programming along a *sphere-cut decomposition*—a type of branch decomposition specifically designed for planar graphs of small treewidth [18]. However, unlike in previous applications of sphere-cut decompositions [39, 44], our algorithm requires the nooses delimiting the bags in the sphere-cut decomposition to admit a fixed drawing since our arguments rely on constructing a hypothetical solution (a subhamiltonian curve) that is "well-behaved" w.r.t. a fixed set of curves. While this would typically lead to extensive case analysis to compute the records of a parent noose from the records of the children, we introduce a generic framework that allows us to transfer records from child to parent nooses via XOR operations. We believe that this may be of broader interest, especially when working with problems which require one to enhance the embedding or drawing of an input graph.

In the final part of the article, we turn our attention to the parameterized complexity of computing book embeddings. While Two-Page Book Embedding is fixed-parameter tractable when parameterized by the treewidth of the input graph, the only graph parameter which has been shown to yield fixed-parameter algorithms for computing ℓ-page book embeddings for $\ell > 2$ is the *vertex cover number*[1] [7]. Whether this tractability result also holds for other structural graph parameters such as treewidth, *treedepth* [46] or the *feedback edge number* [51] has been stated as an open question in the field[2]. We conclude by providing a novel fixed-parameter algorithm for computing ℓ-page book embeddings (or determining that one does not exist) under the third parameterization mentioned above—the feedback edge number, i.e., the edge deletion distance to acyclicity. This result is complementary to the known vertex-cover based fixed-parameter algorithm, and can be seen as a necessary stepping stone towards eventually settling the complexity of computing ℓ-page book embeddings parameterized by treewidth. Moreover, since the obtained kernel is linear in the case of $\ell = 2$, the obtained kernel allows us to generalize our main algorithmic result to a run-time of $2^{O(\sqrt{k})} \cdot n^{O(1)}$ where k is the feedback edge number of the input graph.

2 Preliminaries

Basic Notions. We use basic terminology for graphs and multi-graphs [17], and assume familiarity with the basic notions of parameterized complexity and fixed-parameter tractability [13, 19]. The *feedback edge number* of G, denoted by $\mathsf{fen}(G)$, is the minimum size of any *feedback edge set* of G, *i.e.*, a set $F \subseteq E(G)$ such that $G - F = (V(G), E(G) \setminus F)$ is acyclic.

Fact 1. *Let G be a graph. Then, a minimum feedback edge set of G can be computed in time $O(|V(G)| + |E(G)|)$.*

Proof. The theorem follows because any minimum feedback edge set is equal to $E(G) - E(F)$, where F is a spanning forest of G, together with the fact that we can compute a spanning forest of G in time $O(|V(G)| + |E(G)|)$. □

A *cut vertex* of a multi-graph is a vertex whose removal increases the number of connected components. A connected multi-graph that has no cut vertex is called *biconnected*.

We say that a multi-graph G is *planar* if it admits a *planar drawing*, *i.e.*, a drawing in the plane in such a way that its edges are drawn as simple curves which pairwise intersect only at their endpoints. Let D be a planar drawing of G and f a face of D. We denote by $V(f)$ ($E(f)$) all vertices (edges) of G incident with f.

Every face of a connected planar graph equipped with a drawing induces a cyclic sequence $\sigma(f)$ of the vertices in $V(f)$, *i.e.*, the cyclic sequence is obtained by traversing the closed curve representing the border of f in a clock-wise manner. Note that while $\sigma(f)$ can repeat vertices, this is no longer the case if the graph is biconnected, in which case $\sigma(f)$ induces a cyclic order of the vertices in $V(f)$. For convenience, we will represent cyclic orders by sequences; note that each cyclic order on n elements can be equivalently represented by one of n sequences (one for each starting element). For instance, the cyclic orders represented by the sequences $(a_1, \ldots, a_\ell)$ and $(a_i, \ldots, a_\ell, a_1, \ldots, a_{i-1})$ are the same for every i with $1 \leq i \leq \ell$.

The following basic observations about planar graphs will be useful later:

[1]The vertex cover number is the minimum size of a vertex cover, and represents a much stronger restriction on the structure of the input graphs than, e.g., treewidth.
[2]E.g., at **Advances in Parameterized Graph Algorithms** (Spain, May 2–7 2022) and also at Dagstuhl seminar 21293 **Parameterized Complexity in Graph Drawing** [24].

Observation 2. *Let G be a graph with planar drawing D and let f be a face of D. Then, we can draw a simple curve inside f between any two distinct points in f or its border. Moreover, if G is connected and $\sigma(f) = (v_1, \dots, v_\ell)$, then drawing a curve inside f between v_i and v_j with $i < j$ splits f into two faces f_1 and f_2 such that $\sigma(f_1) = (v_i, \dots, v_j)$, $\sigma(f_2) = (v_j, \dots, v_i)$.*

Observation 3. *Let f be a face of a planar drawing D of a connected graph G and let (v_1, v_2, v_3, v_4) be a subsequence of $\sigma(f)$ such that $v_i \neq v_j$ for every distinct i and j. Then every v_1-v_3 path must intersect every v_2-v_4 path in G in at least one vertex.*

Book Embeddings and Subhamiltonicity. An ℓ-page book embedding of a multi-graph $G = (V, E)$ will be denoted by a pair $\langle \prec, \sigma \rangle$, where $\prec$ is a linear order of V, and $\sigma \colon E \to [\ell]$ is a function that maps each edge of E to one of ℓ pages $[\ell] = \{1, 2, \dots, \ell\}$. In an ℓ-page book embedding $\langle \prec, \sigma \rangle$ it is required that for no pair of edges $uv, wx \in E$ with $\sigma(uv) = \sigma(wx)$ the vertices are ordered as $u \prec w \prec v \prec x$, i.e., each page must be crossing-free. The *page number* of a graph G is the minimum number ℓ such that G admits an ℓ-page book embedding. The general problem of computing the page number of an input graph is thus:

Book Thickness	
Instance:	A multi-graph G with n vertices and a positive integer ℓ.
Question:	Does G admit a ℓ-page book embedding?

It is known that a multi-graph admits a 2-page book embedding if and only if it is *subhamiltonian*, i.e., if it has a planar Hamiltonian supergraph [6]; see Figure 1 for an illustration. It is also known (and also easy to observe) that if G is subhamiltonian, then it has a planar Hamiltonian supergraph G' with $V(G') = V(G)$ and $E(G') \setminus E(G) = E(H)$, where H is a Hamiltonian cycle in G' (see, *e.g.*, [34]). Hence, the problem of deciding whether a graph has page number 2 can be equivalently stated as:

Subhamiltonicity (SUBHAM)	
Instance:	A multi-graph G with n vertices.
Question:	Is G subhamiltonian?

Since the transformation between 2-page book embeddings and Hamiltonian cycles of supergraphs is constructive in both directions, a constructive algorithm for SUBHAM (such as the one presented here) allows us to also output a 2-page book embedding for the graph.

Let G be subhamiltonian. For a Hamiltonian cycle H on $V(G)$ (where H is not necessarily a subgraph of G), we denote by G_H the graph obtained from G after adding the edges of H and say that H is a *witness* for G if G_H is planar. A drawing D of G *respects* H if D can be completed to a planar drawing of G_H by only adding the edges of H. We extend the notion of "witness" to include all the information defining the solution as follows: a tuple (D, D_H, G_H, H) is a *witness* for G if G_H is a planar supergraph of G containing the Hamiltonian cycle H, D_H is a planar drawing of G_H, and D is the restriction of D_H to G; note that D_H witnesses that D respects H.

The following basic observations will be useful when dealing with subhamiltonian graphs in Section 4.1.

Observation 4. *Let G be a subhamiltonian graph with witness (D, D_H, G_H, H). Then:*

(1) Every subgraph of G is also subhamiltonian.

(2) If $uv \in E(H)$, then the graph obtained from G by adding a new vertex x together with the edges xu and xv is subhamiltonian.

(3) If $uv \in E(H)$, then the graph obtained by contracting uv is also subhamiltonian.

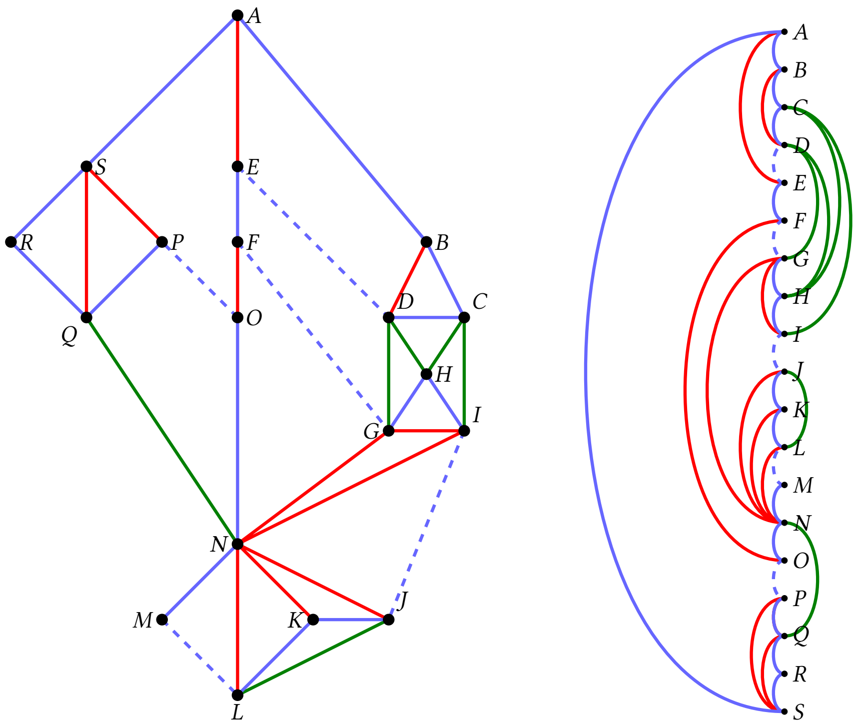

Fig. 1. A drawing of a subhamiltonian graph G, made of the full-edges, which is completed by the dashed edges to one of its Hamiltonian supergraphs G_H (left) and the same graph drawn as a two-page book embedding (right). In both drawings the Hamiltonian cycle H is colored in blue and the edges belonging to page 1 and 2 are colored with green and red, respectively.Note that the partition of the edges into the pages can be obtained from a planar drawing of G_H by partitioning the edges according to the two regions given by H.

SPQR-Trees. We give a brief introduction to the SPQR-tree data structure for biconnected multi-graphs, following the formalism used by Gutwenger et al. [30]. Let $G = (V, E)$ be a biconnected multi-graph and $a, b \in V$. We can partition E into equivalence classes $E_1, \ldots, E_k$ in the following way: for any two edges $e, e' \in E$, e and e' belong to the same equivalence class if and only if there exists a path P in G which contains both e and e' as edges and no internal vertex of P is in $\{a, b\}$. The classes E_i are called the *separation classes* of G with respect to $\{a, b\}$ and if $k \geq 2$ then $\{a, b\}$ is called a *separation pair* unless (i) $k = 2$ and one of the separation classes only contains a single edge, or (ii) $k = 3$ and each separation class is made of a single edge. A biconnected multi-graph without a separation pair is called *triconnected.* A *split pair* is a pair of vertices which are either adjacent to each other, or form a separation pair.

SPQR-trees were introduced by Di Battista and Tamassia [4], based on the ideas of Bienstock and Monma [8, 9], and since then have been used in various graph drawing applications, for a survey we refer to the work of Mutzel [45].

SPQR-trees represent the decomposition of a biconnected multi-graph G based on split pairs and their "split components". A *split component* of a split pair $\{u, v\}$ is either the edge (u, v) or a maximal subgraph C of G such that $\{u, v\}$ is not a split pair of C. Let $\{s, t\}$ be a split pair of G. A *maximal split pair* $\{u, v\}$ of G with respect to $\{s, t\}$ is such that, for any other split pair $\{u', v'\}$, vertices u, v, s and t are in the same split component.

Let $e = (s, t)$ be an edge of G, called the *reference edge*. The SPQR-tree $\mathcal{B}$ of G with respect to e is a rooted ordered tree whose nodes are of four types: S, P, Q, and R. Each node b of $\mathcal{B}$ has an associated biconnected multi-graph $\textsc{Sk}(b)$, called the *skeleton* of b. The tree $\mathcal{B}$ is recursively defined as follows:

- *Trivial Case.* If G consists of exactly two parallel edges between s and t, then $\mathcal{B}$ consists of a single Q-node whose skeleton is G itself.
- *Parallel Case.* If the split pair $\{s, t\}$ has k split components $G_1, \dots, G_k$ with $k \geq 3$, the root of $\mathcal{B}$ is a P-node b, whose skeleton consists of k parallel edges $e = e_1, \dots, e_k$ between s and t.
- *Series Case*: Otherwise, the split pair $\{s, t\}$ has exactly two split components, one of them is e, and the other one is denoted with G'. If G' has cutvertices $c_1, \dots, c_{k-1}$ ($k \geq 2$) that partition G into its blocks $G_1, \dots, G_k$, in this order from s to t, the root of $\mathcal{B}$ is an S-node b, whose skeleton is the cycle $e_0, e_1, \dots, e_k$, where $e_0 = e$, $c_0 = s$, $c_k = t$, and $e_i = (c_{i-1}, c_1)$ ($i = 1, \dots, k$).
- *Rigid Case*: If none of the above cases applies, let $\{s_1, t_1\}, \dots, \{s_k, t_k\}$ be the maximal split pairs of G with respect to $\{s, t\}$ ($k \geq 1$), and, for $i = 1, \dots, k$, let G_i be the union of all the split components of $\{s_i, t_i\}$ but the one containing e. The root of $\mathcal{B}$ is an R-node, whose skeleton is obtained from G by replacing each subgraph G_i with the edge $e_i = (s_i, t_i)$.

Except for the trivial case, b has children $b_1, \dots, b_k$, such that b_i is the root of the SPQR-tree of $G_i \cup e_i$ with respect to e_i ($i = 1, \dots, k$). The endpoints of the edge e_i are called the *poles* of node b_i. Edge e_i is said to be the *virtual edge* of node b_i in the skeleton of b and of node b in the skeleton of b_i. We call node b the *pertinent node* of e_i in the skeleton of b_i, and b_i the *pertinent node* of e_i in the skeleton of b. The virtual edge of b in the skeleton of b_i is called the reference edge of b_i.

Let b_r be the root of $\mathcal{B}$ in the decomposition given above. We add a Q-node representing the reference edge e and make it the parent of b_r so that it becomes the new root.

Let e be an edge in $\textsc{Sk}(b)$ and let b' be the pertinent node of e. Deleting edge $\{b, b'\}$ in $\mathcal{B}$ splits $\mathcal{B}$ into two connected components. Let $\mathcal{B}_{b'}$ be the connected component containing b'. The *expansion graph* of e (denoted with $\textsc{exp}(e)$) is the graph induced by the edges of G contained in the skeletons of the Q-nodes in $\mathcal{B}_{b'}$. We further introduce the notation $\textsc{exp}^+(e)$ for the graph $\textsc{exp}(e) \cup e$. The *pertinent graph* $\textsc{Pe}(b)$ of a tree node b is obtained from $\textsc{Sk}(b)$ minus the reference edge by replacing each skeleton edge with its expansion graph. An illustration of an SPQR-tree is provided in Figure 2.

SPQR-trees can be computed efficiently, and this also implicitly bounds the their size.

Lemma 5 ([29]). *Let G be biconnected multi-graph with n vertices and m edges. An SPQR-tree of G with $O(m)$ nodes and edges inside skeletons can be constructed in $O(n + m)$ time.*

Choosing a different reference edge e' is equivalent to rooting the tree $\mathcal{B}$ at the Q-node whose skeleton contains e'. In particular, the unrooted version of the SPQR-tree of a biconnected multi-graph (including the skeleton graphs) is unique.

We will later also need the following well-known fact about SPQR-trees, which will need to define the types of nodes in an SPQR-tree.

Fact 6. *Let G be a biconnected planar multi-graph with planar drawing D and let $\mathcal{B}$ be the SPQR-tree of G. Then, for every node b of $\mathcal{B}$ with reference edge (s_b, t_b), there is a noose N_b such that:*

- *N_b intersects with D only at s_b and t_b.*
- *N_b separates $\textsc{Pe}(b)$ from $G \setminus \textsc{Pe}(b)$ in D.*

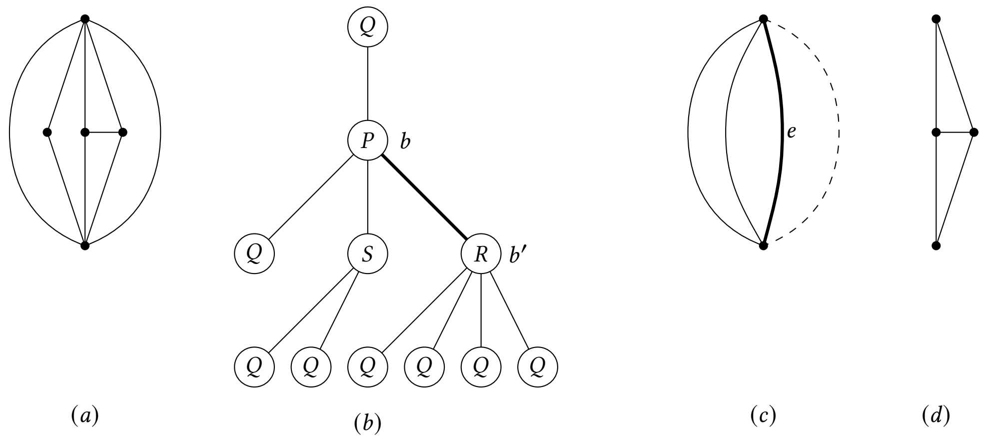

Fig. 2. (a) shows a biconnected multi-graph G. (b) shows the SPQR-tree $\mathcal{B}$ of G. (c) shows the skeleton of b, $\textsc{Sk}(b)$, where the edge e that corresponds to the child (with pertinent node) b' is in bold and the dashed edge represents the reference edge. Finally, (d) shows $\textsc{Pe}(b')$.

Moreover, if N_b and $N_{b'}$ for two nodes b and b' of $\mathcal{B}$ have the same reference edge (s,t) and contain a subcurve between s and t in the same face of D, then we can and will assume that the two subcurves are identical.

Sphere-Cut Decompositions. A branch decomposition $\langle T, \lambda\rangle$ of a graph G consists of an unrooted ternary tree T (meaning that each node of T has degree one or three) and of a bijection $\lambda : \mathcal{L}(T) \leftrightarrow E(G)$ from the leaf set $\mathcal{L}(T)$ of T to the edge set $E(G)$ of G; to distinguish $E(T)$ from $E(G)$, we call the elements of the former *arcs* (as was also done in previous work [18]). For each arc a of T, let T_1 and T_2 be the two connected components of $T - a$, and, for $i = 1, 2$, let G_i be the subgraph of G that consists of the edges corresponding to the leaves of T_i, i.e., the edge set $\{\lambda(\mu) : \mu \in \mathcal{L}(T) \cap V(T_i)\}$. The middle set $mid(a) \subseteq V(G)$ is the intersection of the vertex sets of G_1 and G_2, i.e., $mid(a) := V(G_1) \cap V(G_2)$. The width $\beta(\langle T, \lambda\rangle)$ of $\langle T, \lambda\rangle$ is the maximum size of the middle sets over all arcs of T, i.e., $\beta(\langle T, \lambda\rangle) := max\{|mid(a)| : a \in E(T)\}$. An optimal branch decomposition of G is a branch decomposition with minimum width; this width is called the branchwidth $\beta(G)$ of G. We will need the following well-known relation between treewidth and branchwidth.

Lemma 7 ([48, Theorem 5.1])**.** *Let G be a graph. Then, $bw(G) - 1 \le tw(G) \le \frac{3}{2}bw(G) - 1$, where $bw(G)$ is the branchwidth and $tw(G)$ is the treewidth of G.*

Let D be a plane drawing of a connected planar graph G. A noose of D is a closed simple curve that (i) intersects D only at vertices and (ii) traverses each face at most once, i.e., its intersection with the region of each face forms a connected curve. The length of a noose is the number of vertices it intersects, and every noose O separates the plane into two regions δ_1 and δ_2. A *sphere-cut decomposition* $\langle T, \lambda, \Pi = \{ \pi_a \mid a \in E(T) \}\rangle$ of (G, D) is a branch decomposition$\langle T, \lambda\rangle$ of G together with a set Π of circular orders π_a of $mid(a)$—one for each arc a of T—such that there exists a noose O_a whose closed discs δ_1 and δ_2 enclose the drawing of G_1 and of G_2, respectively. Observe that O_a intersect G exactly at $mid(a)$ and its length is $|mid(a)|$. Note that the fact that G is connected together with Conditions (i) and (ii) of the definition of a noose implies that the graphs G_1 and G_2 are both connected and that the set of nooses forms a laminar set family, that is, any two nooses are either disjoint or nested. A clockwise traversal of O_a in the drawing of G defines the cyclic ordering π_a of $mid(a)$. We always assume that the vertices of every middle set $mid(a)$ are enumerated according to π_a. A sphere-cut decomposition of a given planar graph with n vertices can be constructed in $O(n^3)$ time [18].

Lemma 8 ([18, Theorem 1])**.** *Let G be a biconnected planar multi-graph on n vertices and branchwidth ω. Then, a sphere-cut decomposition of G of width ω can be computed in time $O(n^3)$.*

Note that [18, Theorem 1] requires that G has not vertices of degree at most one, which is the case for biconnected multi-graphs.

We will only consider sphere-cut decompositions of $\textsc{Sk}(b)$ for some R-node or S-node b in an SPQR-tree, which implies that the underlying graph will admit a unique planar embedding. Due to this fact, we sometimes abuse the notation by referring to sphere-cut decompositions as purely combinatorial objects (i.e., without an explicit drawing of the individual nooses). Suppose that b is an R-node in some SPQR-tree and let $\langle T_b, \lambda_b, \Pi_b \rangle$ be a sphere-cut decomposition for graph $\textsc{Sk}(b)$ with the reference edge $\{(s_b, t_b)\}$. Let ${\lambda_b}^{(-1)}((s_b, t_b))$ be the root of T_b. Each arc a of T_b is associated with the subgraph $\textsc{Sk}(b, a)$ of $\textsc{Sk}(b) \cup \{(s_b, t_b)\}$ in the inside region, i.e., the region not containing the reference edge, of the noose O_a of a. The *pertinent graph* $\textsc{Pe}(b, a)$ of an R-node b is obtained from $\textsc{Sk}(b, a)$ by replacing each skeleton edge with its expansion graph.

Every noose O_a can be divided into subcurves by splitting the noose at the vertices in $mid(a)$. Each such subcurve can be characterized by a pair $(\{u, v\}, f)$, where $u, v \in mid(a)$ are two consecutive nodes in π_a and f is a face of $\textsc{Sk}(b) \cup \{(s_b, t_b)\}$. Due to the properties of sphere-cut decompositions, we can assume that whenever two nooses contain two subcurves that are characterized by the same pair, then the subcurves are identical. For convenience, we can identify any noose of the sphere-cut decomposition with the set of subcurves that it contains, e.g., we often view the noose O_a as the set of pairs $(\{u, v\}, f)$ that correspond to the subcurves contained in O_a.

Below, we note that the notion of nooses defined above can also be assumed to be well-behaved when dealing with sphere-cut decompositions of an R-node or an S-node in an SPQR-tree of G.

Observation 9. *Let G be a biconnected planar multi-graph with planar drawing D, let $\mathcal{B}$ be the SPQR-tree of G and let b be an R-node or an S-node of $\mathcal{B}$ with sphere-cut decomposition $\langle T_b, \lambda_b, \Pi_b \rangle$. Then, D can be extended to a planar drawing of G together with the nooses $\{O_a \mid a \in E(T_b)\}$ where each of the nooses lies inside N_b.*

We say that a biconnected planar multi-graph G equipped with an SPQR-tree $\mathcal{B}$ is *associated* with a set $\mathcal{T}$ of sphere-cut decompositions if $\mathcal{T}$ contains a sphere-cut decomposition of $\textsc{Sk}(b)$ for every R-node and every S-node b of $\mathcal{B}$. The following lemma now follows immediately from Fact 6 and Observation 9.

Lemma 10. *Let G be biconnected planar multi-graph with planar drawing D and SPQR-tree $\mathcal{B}$ of G together with the associated set $\mathcal{T}$ of sphere-cut decompositions. Then, D can be extended to a planar drawing D' of G together with all nooses in $\{\, O_a \mid a \in E(T_b) \wedge \langle T_b, \lambda_b, \Pi_b \rangle \in \mathcal{T} \,\}$ as well as a noose N_b for every node b of $\mathcal{B}$ satisfying:*

- *N_b intersects with D only at s_b and t_b.*
- *N_b separates $\textsc{Pe}(b)$ from $G \setminus \textsc{Pe}(b)$ in D.*

Moreover, if any of the subcurves of the nooses O_a and the nooses N_b connect the same two vertices in the same face of D, then the two subcurves are identical in D'.

Non-Crossing Matchings. We will use non-crossing matchings and the closely related Dyck words for the definition and analysis of our types. Let K_n be the complete graph on vertices $\{1, \ldots, n\}$ and let $<$ be a cyclic ordering of the elements in $\{1, \ldots, n\}$. A *non-crossing matching* is a matching M in the graph K_n such that for every two edges $\{a, b\}, \{c, d\} \in M$

it is not the case that $a < c < b < d$. A non-crossing matching can be visualized by placing n vertices on a cycle, and connecting matching vertices with pairwise non-crossing curves all on one fixed side of the cycle. The number of non-crossing matchings over n vertices is given by [41, 50]:

$$M(n) = CN(\frac{n}{2}) \approx \frac{2^n}{\sqrt{\pi}(\frac{n}{2})^{\frac{3}{2}}} \approx 2^n$$

Here, $CN(n)$ is the n-th Catalan number, i.e.:

$$CN(n) = \frac{1}{n+1}\binom{2n}{n} \approx \frac{4^n}{\sqrt{\pi}n^{\frac{3}{2}}} \approx 4^n$$

A *Dyck word* is a sequence composed of {"[", "]"} symbols, such that each prefix has an equal or greater number of "["s than "]"s, and the total number of "["s and "]"s are equal.

Observation 11. *There is a one-to-one correspondence between non-crossing matchings with* $2n$ *vertices and Dyck words of length* $2n$*. Moreover, one can be translated into the other after fixing a starting vertex and an orientation of the cycle.*

3 Solution Normal Form

Our first order of business is to show that we can assume that the solution (Hamiltonian cycle) to the SUBHAM problem interacts with the drawing in a restricted manner. In particular, we aim to show that every subhamiltonian graph G has a witness (D, D_H, G_H, H) in *normal form*, i.e., with the following property: it is possible to draw a curve in D_H between any two vertices occurring in a common face of D such that this curve only crosses the Hamiltonian cycle at most twice. Note that this property will allow us to bound the number of possible interactions of the Hamiltonian cycle with any subgraph corresponding to either a node in the SQPR-tree or an arc in a sphere-cut decomposition and is crucial to bound the number of types in our dynamic programming algorithm.

We will need the following auxiliary lemmas.

Lemma 12. *Let* G *be a subhamiltonian graph with witness* (D, D_H, G_H, H) *and let* f *be a face of* D_H*. Then the restriction of the cyclic order given by* H *to the vertices in* $V(f)$ *is either equal to* $\sigma(f)$ *or it is equal to the reverse of* $\sigma(f)$*.*

Proof. Suppose for a contradiction that the cyclic order of $\sigma(f)$ differs from the (reverse) cyclic order given by H. Then, there are three vertices a, b, and c such that b is between a and c in the cyclic order given by $\sigma(f)$, but b is between d and e in the cyclic order given by H, where $\{d, e\} \neq \{a, c\}$. W.l.o.g. assume that $d \neq a$. Then, H contains a path P_{db} between d and b that does not contain any vertex from $V(f) \setminus \{d, b\}$ and moreover d is neither between a and b nor between b and c in the cyclic order given by $\sigma(f)$. Let A be the set of all vertices between d and b in the cyclic order defined by $\sigma(f)$. Then, because H is a Hamiltonian cycle and $V(f) \setminus A \neq \emptyset$, we obtain that there is a vertex $x \in A$ and a vertex $y \in V(f) \setminus A$ such that H contains a path P_{xy} that does not contain any vertex in $V(f) \setminus \{x, y\}$. Since P_{db} and P_{xy} are disjoint, the statement of the lemma now follows from Observation 3. □

Lemma 13. *Let* G *be a subhamiltonian graph with witness* (D, D_H, G_H, H) *and let* f *be a face of* D*. For any two vertices* $u, v \in V(f)$*, a* uv*-curve* c *can be added to* D_H *inside* f *such that every edge from* $E(H)$ *crosses at most once with* c*.*

Proof. Let D'_H be obtained from the restriction of D_H to vertices and edges inside f and let f_u and f_v be the two faces of D'_H inside f having u or v on their border, respectively. If $f_u = f_v$, then the claim follows immediately from Observation 2. Otherwise, consider the dual graph H of D'_H together with its drawing D^D_H inside D'_H. Then, H contains a path from f_u to f_v that uses only faces inside f and that corresponds to a curve P between f_u and f_v in D_H^D that intersects every edge of H at most once. Because of Observation 2, we can draw a curve c_u from u to f_u and a curve c_v from f_v to v inside f_u and f_v, respectively, without adding any crossings. Then, the curve obtained from the concatination of c_u, P, and c_v is the required uv-curve. □

Lemma 14. *Let G be a subhamiltonian graph with witness (D, D_H, G_H, H), let f be a face of D, and let c be a curve drawn inside f between two vertices $u, v \in V(f)$. Then, we can redraw the curves corresponding to the edges of H inside f such that every such curve crosses c at most once, i.e., we can adapt D_H inside f into a drawing D'_H such that (D, D'_H, G_H, H) is a witness for G and every curve corresponding to an edge of H inside f crosses c at most once in D'_H.*

Proof. Because of Lemma 13 there is a uv-curve c' that can be added to D_H inside f such that every curve corresponding to a Hamiltonian cycle of H crosses c' at most once. Let $(p_1, \ldots, p_\ell)$ be the sequence of all crossing points between c' and curves corresponding to edges of H given in the order of appearance when going along c' from u to v and suppose that p_i is the crossing point of the edge e_i on H with c'.

Now consider the drawing D_H^- obtained from D_H after adding c and removing all curves corresponding to edges of H inside f. Moreover, let $p_1^c, \ldots, p_\ell^c$ be an arbitrary set of pairwise distinct points on c that occur in the order $(p_1^c, \ldots, p_\ell^c)$ when going along c from u to v and let c^1 and c^2 be the two subcurves in D_H^- of the border of f between u and v. Note that every edge e_i has one endpoint v_i^1 on c^1 and one endpoint v_i^2 on c^2; otherwise both endpoints of e_i are either on c^1 or on c^2 and c' could have been drawn in D_H without crossing the curve corresponding to e_i. Note furthermore that because of Observation 3, the vertices $v_1^j, \ldots, v_\ell^j$ must appear in the order $(v_1^j, \ldots, v_\ell^j)$, when going along c^j from u to v for every $j \in \{1, 2\}$. Since v_i^1 (v_i^2) and p_i are initially in the same face of D_H^-, we can use Observation 2 to draw a curve between v_i^1 (v_i^2) and p_i in this face for every $i \in [1, \ell]$. Moreover, using the same observation, we obtain that after drawing this curve, it still holds that v_i^1 (v_i^2) are in the same face as p_i for every $i \in [1, \ell]$. Therefore, we can repeatedly apply Observation 2 to draw curves in D_H^- between v_i^j and p'_i for every $i \in [1, \ell]$ and $j \in \{1, 2\}$ to obtain the required drawing D'_H. □

The following lemma is crucial to obtain our normal form. An illustration of the main ideas behind the proof is provided in Figure 3.

Lemma 15. *Let G be a subhamiltonian graph with witness (D, D_H, G_H, H), let f be a face of D and let c be a curve drawn inside f between two vertices $u, v \in V(f)$. Then, there is a witness $(D, D_{H'}, G_{H'}, H')$ for G such that:*

(1) $D_{H'}$ and D_H differ only inside f.
(2) c crosses at most two curves corresponding to the edges of H'.
(3) c crosses each curve corresponding to an edge of H' at most once.

Proof. By Lemma 14, we can assume that the every curve corresponding to an edge of H inside f crosses c at most once in D_H, which shows *(3)*. If c crosses at most two edges of H, then the statement of the lemma holds. So suppose that this is not the case and let u_1v_1, u_2v_2, and u_3v_3 be three distinct edges in $E(H) \setminus E(G)$ that cross c at three successive points p_1, p_2, and p_3 such that no other edge of $E(H) \setminus E(G)$ crosses c between p_1 and p_3. Assume furthermore that u_1, u_2, and u_3 are on the same face in $D + c$, where here and in the following $D + c$ denotes the drawing obtained from D after adding c, and the same for v_1, v_2, and v_3. Then, there are faces f_H^1 and f_H^3 of D_H such that $u_1, v_1, u_2, v_2 \in V(f_H^1)$ and $u_2, v_2, u_3, v_3 \in V(f_H^3)$.

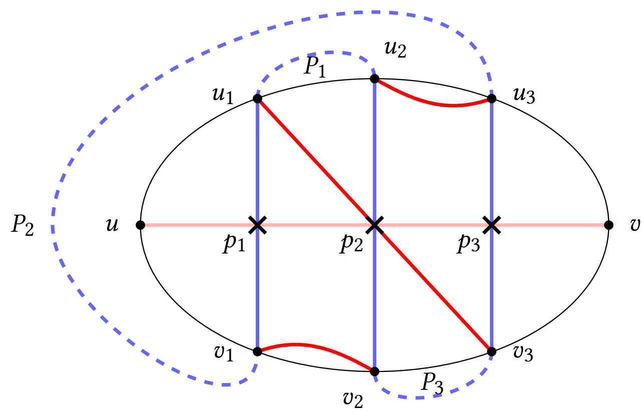

Fig. 3. The cycle $H = (u_2, P_1, u_1, v_1, P_2, u_3, v_3, P_3, v_2, u_2)$ represents a Hamiltonian cycle that crosses the uv-curve at least three times (in p_1, p_2 and p_3). Thanks to Lemma 15, we obtain a Hamiltonian cycle $H' = (u_2, P_1, u_1, v_3, P_3, v_2, v_1, P_2, u_3, u_2)$ that differs from H only inside the face $f = (u, u_1, u_2, u_3, v, v_3, v_2, v_1)$ and crosses the uv-curve two fewer times than H does. Finally, note that the vertices u and v are part of either P_1, P_2, or P_3.

Now we will analyze the Hamiltonian cycle H. For every i in $\{1, 2, 3\}$, the edge u_iv_i is in $E(H)$. Let P_H be the path from u_2 to v_2 obtained by deleting the edge u_2v_2 from H. From Observation 12 applied to f_H^i we obtain that u_i is between u_2 and v_i in P_H, for i in $\{1, 3\}$. Let P_1, P_2, P_3 be paths created after deleting u_1v_1 , u_2v_2, u_3v_3 from H. Then, either:

- $P_H = (u_2, P_1, u_1, v_1, P_2, u_3, v_3, P_3, v_2)$ or
- $P_H = (u_2, P_1, u_3, v_3, P_2, u_1, v_1, P_3, v_2)$.

The proof for both cases is entirely analogous, so we will only make the first case explicit. An illustration of the current setting is provided in Figure 3.

$H' = (V(H), (E(H) \setminus \{u_1v_1, u_2v_2, u_3v_3\}) \cup \{u_1v_3, v_2v_1, u_3u_2\}$ is a Hamiltonian cycle, because it corresponds to the sequence $(u_2, P_1, u_1, v_3, P_3, v_2, v_1, P_2, u_3)$. At this point, we have to prove that there exists a planar drawing $D_{H'}$ of $G_{H'}$ that satisfies *(1)–(3)*. To do so we will change D_H.

Let $G_H^\star$ be the graph obtained from G_H after subdividing the edges u_iv_i with the new vertex p_i for every i in $\{1, 2, 3\}$ and adding the edges p_1p_2 and p_2p_3. Note that $G_H^\star$ is planar, as witnessed by the drawing $D^\star = D_H + c'$, where c' is the restriction of c to the segment between p_1 and p_3.

Because u_2 and u_3 lie on the same face as p_2 and p_3 in $D^\star$, we obtain from Observation 2 that we can add the curve between u_2 and u_3 inside this face without adding any crossings. Analogously, we will add the curves v_1v_2, u_1p_2 and p_2v_3. We can now obtain a new drawing D' from $D^\star$ by removing the curves u_iv_i and adding the curves u_2u_3, v_1v_2, and the curve u_1v_3 obtained as the concatenation of the curves u_1p_2 and p_2v_3.

Observe that in this new drawing we reduced the number of crossings by 2, *i.e.*, instead of the crossings at p_1, p_2, and p_3, only the crossing at p_2 remains (u_1v_3-curve). Moreover, all changes happened inside f.

Finally, let $D_{H'}$ be the drawing obtained from D' after removing the curve between u and v. Then, $D_{H'}$ shows *(1)*. Moreover, by repeating the process as long as we have at least 3 crossings with the uv-curve, we obtain a drawing that also satisfies *(2)*. ◻

We are now ready to define our normal form for the Hamiltonian cycle. Essentially, we show that if there is a Hamiltonian cycle, then there is one which crosses each subcurve that is either part of the border of a node in the SPQR-tree or that is a subcurve of some noose in a sphere-cut decomposition of an R-node or an S-node at most twice.

Let G be a biconnected subhamiltonian multi-graph with SPQR-tree $\mathcal{B}$ and the associated set $\mathcal{T}$ of sphere-cut decompositions $\langle T_b, \lambda_b, \Pi_b \rangle$ of $\textsc{Sk}(b)$ for every R-node and S-node b of $\mathcal{B}$. We say that a witness $W = (D, D_H, G_H, H)$ for G *respects* the sphere-cut decompositions in $\mathcal{T}$, if there is a planar drawing of all nooses in the sphere-cut decompositions of $\mathcal{T}$ into D such that every subcurve c in $\bigcup_{a \in E(T_b)} O_a$ crosses the curves corresponding to the edges of H at most twice in D_H. We say that the witness W for G *respects* $\mathcal{B}$ if it respects the sphere-cut decompositions in $\mathcal{T}$ and for every node b of $\mathcal{B}$ with reference edge (s_b, t_b), it holds that there is a noose N_b that can be drawn into D_H such that:

- N_b touches D only at s_b and t_b.
- N_b separates $\textsc{Pe}(b)$ from $G \setminus \textsc{Pe}(b)$ in D.
- Each of the two subcurves L_b and R_b obtained from N_b by splitting N_b at s_b and t_b crosses the curves corresponding to the edges of H at most twice.
- Moreover, if any of the subcurves of the nooses O_a and the nooses N_b connect the same two vertices in the same face of D, then the two subcurves are identical.

The following lemma allows us to assume our normal form and follows easily from Lemma 10 together with a repeated application of Lemma 15.

Lemma 16. *Let G be a biconnected subhamiltonian multi-graph with SPQR-tree $\mathcal{B}$ and the associated set $\mathcal{T}$ of sphere-cut decompositions. Then, there is a witness $W = (D, D_H, G_H, H)$ for G that respects $\mathcal{B}$.*

Proof. Let $W = (D, D_H, G_H, H)$ be any witness for G, which exists because G is subhamiltonian. Let D' be the planar drawing obtained from D using Corollary 10. That is, D' is a planar drawing of G together with all nooses in $\{ O_a \mid a \in E(T_b) \wedge \langle T_b, \lambda_b, \Pi_b \rangle \in \mathcal{T} \}$ as well as a noose N_b for every node b of $\mathcal{B}$ satisfying:

- N_b intersects with D only at s_b and t_b.
- N_b separates $\textsc{Pe}(b)$ from $G \setminus \textsc{Pe}(b)$ in D.

Moreover, if any of the subcurves of the nooses O_a and the nooses N_b connect the same two vertices in the same face of D, then the two subcurves are identical in D'. Note that this implies that every face of D contains at most one subcurve from the nooses O_a and N_b.

Similarly, let D'_H be obtained in the same manner from D_H. If W already respects $\mathcal{B}$, then there is nothing to show. Otherwise, it holds that either:

- there is a noose N_b such that L_b or R_b are crossed by the curves corresponding to the edges of H more than twice or
- there is a subcurve $c \in O_a$ for some $a \in E(T_b)$ and $\langle T_b, \lambda_b, \Pi_b \rangle \in \mathcal{T}$ that is crossed by the curves corresponding to the edges of H more than twice

Since every face of D contains at most one subcurve from the nooses O_a and N_b, it follows that in both cases, we can apply Lemma 15 to obtain a witness W' for G that crosses L_b, R_b, or c, respectively, at most twice and does not introduce any additional crossings. This implies that a repeated application of Lemma 15 allows us to obtain the desired witness that crosses each of the subcurves added to D_H in D'_H at most twice. □

4 Setting Up the Framework

In this section we provide the foundations for our algorithm. That is, in Subsection 4.1, we show that it suffices to consider biconnected graphs allowing us to employ SPQR-trees. We then define the types for nodes in the SPQR-tree, which we compute in our dynamic programming algorithm on SPQR-trees, in Subsection 4.2. Finally, in Subsection 4.3 we introduce our general framework for simplifying dynamic programming algorithms on sphere-cut decompositions and introduce the types for nodes of a sphere-cut decompositionthat we compute as part of our dynamic programming algorithm on sphere-cut decompositions.

4.1 Reducing to the Biconnected Case

We begin by showing that any instance of SUBHAM can be easily reduced to solving the same problem on the biconnected components of the same instance. It is well-known that SUBHAM can be solved independently on each connected component of the input graph, the following theorem now also shows that the same holds for the biconnected components of the graph and allows us to employ SPQR-trees for our algorithm.

Theorem 17. *Let G be a graph and let $C \subseteq V(G)$ such that $N(C) = \{n\}$, where $N(C) = \{\, v \in V(G) \setminus C \mid \exists c \in C\ \{v, c\} \in E(G)\,\}$ is the set of neighbors of any vertex of C in $V(G) \setminus C$. Then G is subhamiltonian if and only if both $G^- = G - C$ and $G^C = G[C \cup \{n\}]$ are subhamiltonian.*

Proof. If G is subhamiltonian, then because G^- and G^C are both subgraphs of G we obtain from Observation 4 *(1)* that G^- and G^C are also subhamiltonian.

Towards showing the reverse direction, suppose that G^- and G^C are subhamiltonian. Therefore, G^- and G^C have witnesses $(D^-, D_{H^-}, G_{H^-}, H^-)$ and $(D^C, D_{H^C}, G_{H^C}, H^C)$, respectively. Let $e^- = n^-v^-$ and $e^C = n^Cv^C$ be one of the two edges incident to n in H^- and H^C, respectively. Because any face can be drawn as the outer face of a planar graph, we can assume w.l.o.g. that the edges e^- and e^C are incident to the outer faces of the drawings D_{H^-} and D_{H^C}, respectively.

Let G' be the graph obtained via the disjoint union of G^- and G^C. Then, G' is subhamiltonian because the cycle $H' = (V(G'), (E(H^-) \cup E(H^C) \cup \{n^-n^C, v^-v^C\}) \setminus \{e^-, e^C\})$ is a Hamiltonian cycle of G' that has a planar drawing $D_{H'}$ which is obtained from the disjoint union of the drawing D_{H^-} and D_{H^C} after adding the edges n^-n^C and v^-v^C using Observation 2. Then from Observation 4 *(3)* applied to G' for the edge $n^-n^C \in E(H')$, we conclude that G is also subhamiltonian, as desired. □

4.2 Defining the Types for Nodes in the SPQR-tree

Here, we define the types for nodes in the SPQR-tree that we will later compute using dynamic programming. In the following, we assume that G is a biconnected multi-graph with SPQR-tree $\mathcal{B}$ and the associated set $\mathcal{T}$ of sphere-cut decompositions. Let b be a node of $\mathcal{B}$ with pertinent graph $\text{Pe}(b)$ and reference edge $e = (s, t)$. A *type* of b is a triple (ψ, M, S) such that (please refer also to Figure 4 for an illustration of some types):

- ψ is a function from $\{L, R\}$ to subsets of $\{l, l', r, r'\}$ such that $\psi(L) \in \{\emptyset, \{l\}, \{l, l'\}\}$ and $\psi(R) \in \{\emptyset, \{r\}, \{r, r'\}\}$. We denote by $V(\psi)$ the set $\psi(L) \cup \psi(R)$. Informally, ψ captures how many times the Hamiltonian cycle enters and exits the graph $\text{Pe}(b)$ from the left (L) and from the right (R).

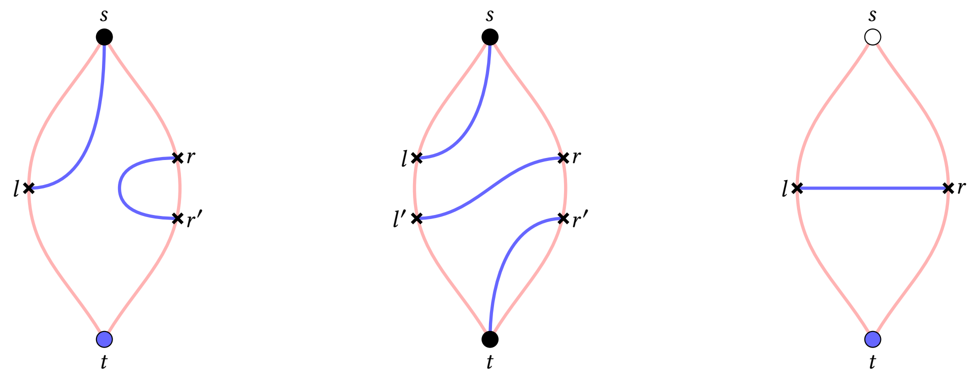

Fig. 4. The figure shows three different types of a node in an SPQR-tree with reference edge (s,t), i.e., the types shown are (from left to right): $(\{\{L \to \{l\}\}, \{R \to \{r,r'\}\}, \{\{l,s\},\{r,r'\}\}, \{t\})$, $(\{\{L \to \{l,l'\}\}, \{R \to \{r,r'\}\}, \{\{l,s\},\{l',r\},\{t,r'\}\}, \emptyset)$, and $(\{\{L \to \{l\}\}, \{R \to \{r\}\}, \{\{l,r\}\}, \{t\})$. The subset of $\{l,l'\}$ and $\{r,r'\}$ that appears corresponds to $\psi(L)$ and $\psi(R)$ respectively. The blue edges correspond to the matching M and the blue vertices corresponds to S.

- $M \subseteq \{\,\{u,v\} \mid u,v \in \{s,t\} \cup V(\psi) \wedge u \neq v\,\}$ and M is a non-crossing matching w.r.t. the circular ordering (s,r,r',t,l',l) that matches all vertices in $V(\psi)$ (i.e. $V(\psi) \subseteq V(M)$), where $V(M) = \bigcup_{e \in M} e$. Informally, M captures the maximal path segments of the Hamiltonian cycle inside $\mathrm{Pe}(b) \cup V(\psi)$ with endpoints in $\{s,t\} \cup V(\psi)$.
- $S \subseteq \{s,t\} \setminus V(M)$. Informally, S captures whether s or t are contained as inner vertices on path segments corresponding to M.

We now provide the formal semantics of types; see Figure 4 for an illustration. Let $\mathcal{X}$ be the set of all types and $\mathrm{Pe}^*(b)$ be the graph obtained from $\mathrm{Pe}(b)$ after adding the dummy vertices l, l', r, and r' together with the edges sl, ll', $l't$, sr, rr', and $r't$. We say that b has type $X = (\psi, M, S)$ if there is a set $\mathcal{P}$ of vertex-disjoint paths or a single cycle in the complete graph with vertex set $V(\mathrm{Pe}^*(b))$ such that:

- $\mathcal{P}$ consists of exactly one path P_e between u and v for every $e = \{u,v\} \in M$ or $\mathcal{P}$ is a cycle and $M = \emptyset$.
- $\{\,\mathrm{IN}(P) \mid P \in \mathcal{P}\,\}$ is a partition of $(V(\mathrm{Pe}(b)) \setminus \{s,t\}) \cup S$, where $\mathrm{IN}(P)$ denotes the set of inner vertices of P.
- there is a planar drawing $D(b,X)$ of $\mathrm{Pe}^*(b) \cup \bigcup_{P \in \mathcal{P}} P$ with outer-face f such that $\sigma(f) = \{s,r,r',t,l',l\}$.

The way we define the types $X = (\psi, M, S)$ of a node b allows us to associate each witness with a type based on the restriction of the witness to the respective pertinent graph.

Formally, let $W = (D, D_H, G_H, H)$ be a witness for G that respects $\mathcal{B}$. Then, the *type* of a node b in the SPQR-tree of G *w.r.t.* W, denoted by $\Gamma_W(b)$, is obtained as follows. Let D_H^b be the drawing D_H restricted to the region with border N_b containing $\mathrm{Pe}(b)$. Let C be the set of all curves in D_H^b corresponding to path segments of H. Let P_L (P_R) be the set of all endpoints of curves in C in L_b (R_b). Note that $|P_L| \leq 2$ and $|P_R| \leq 2$, the endpoints of every curve in C are from the set $\{s,t\} \cup P_L \cup P_R$, and the inner vertices of the curves represent a partition of $V(\mathrm{Pe}(b)) \setminus \{s,t\}$. W.l.o.g., we assume that $P_L \subseteq \{l,l'\}$ and $P_R \subseteq \{r,r'\}$. We are now ready to define the type $X = (\psi, M, S)$ for b. Let M be the set containing the set of endpoints for every curve in C, let $\psi(L) = P_L$ and $\psi(R) = P_R$, and let $S \subseteq \{s,t\}$ be the set that contains s (or t) if s (t) occurs as an inner vertex of some curve in C. Then, the type of b w.r.t. W is equal to (ψ, M, S). Note also that the drawing D_H^b witnesses that b has type (ψ, M, S).

4.3 Framework for Sphere-cut Decomposition

Here, we introduce our framework to simplify the computation of records via bottom-up dynamic programming along a sphere-cut decomposition. Since the framework is independent of the type of records one aims to compute, we believe that the framework is widely applicable and therefore interesting in its own right. In particular, we introduce a simplified framework for computing the types of arcs (or, equivalently, nooses) in sphere-cut decompositions.

Indeed, the central ingredient of any dynamic programming algorithm on sphere-cut decompositions is a procedure that given an inner node with parent arc a_P and child arcs a_L and a_R computes the set of types for the noose O_{a_P} from the set of types for the nooses O_{a_L} and O_{a_R}. Unfortunately, there is no simple way to obtain O_{a_P} from O_{a_L} and O_{a_R} and this is why computing the set of types for O_{a_P} from the set of types for O_{a_L} and O_{a_R} usually involves a technical and cumbersome case distinction [18]. To circumvent this issue, we introduce a simple operation, i.e., the $\oplus$ (**XOR**) operation defined below, and show that the noose O_{a_p} can be obtained from the nooses O_{a_L} and O_{a_R} using merely a short sequence—one of length at most four—of $\oplus$ operations.

Central to our framework is the notion of *weak nooses*, which are defined below and can be seen as intermediate results in the above-mentioned sequence of simple operations from the child nooses to the parent noose; in particular, weak nooses are made up of subcurves of the nooses in the sphere-cut decomposition. Let G be a biconnected multi-graph and let $\mathcal{B}$ be an SPQR-tree of G. Let b be an R-node or S-node of $\mathcal{B}$ with pertinent graph $\textsc{Pe}(b)$. Let $\langle T_b, \lambda_b, \Pi_b \rangle$ be a sphere-cut decomposition of $\textsc{Sk}(b)$ and a be an arc of T_b with pertinent graph $\textsc{Pe}(b, a)$. Let $C(T_b)$ be the set of all subcurves of all nooses occurring in T_b, i.e., $C(T_b) = \bigcup_{a \in E(T_b)} O_a$ where O_a is seen as a set of subcurves. We say O is a *weak noose* if O is a noose consisting only of subcurves from $C(T_b)$.

We now create the equivalent definition of $mid(a)$, which is defined on subsets of subcurves in $C(T_b)$ (and therefore also weak nooses) instead of on arcs in the sphere-cut decomposition, as follows. For $O \subseteq C(T_b)$, the set $V(O)$ is equal to $\bigcup_{(\{u,v\},f) \in O} \{u, v\}$; note that this definition is equivalent to the definition given for arcs, i.e., $mid(a) = V(O_a)$ holds for every arc a of T_b. We also define $\textsc{Pe}(b, O)$ and $\textsc{Sk}(b, O)$ as the subgraph of $\textsc{Pe}(b)$ and $\textsc{Sk}(b)$, respectively, that is contained inside the weak noose O; note that in particular $\textsc{Pe}(b, O_a) = \textsc{Pe}(b, a)$ and $\textsc{Sk}(b, O_a) = \textsc{Sk}(b, a)$ for any arc a of T_b.

While the above definitions are general, in our setting it will be sufficient to restrict our attention to "local" weak nooses consisting of $O_{a_P} \cup O_{a_L} \cup O_{a_R}$, where a_P is a parent arc for arcs a_L and a_R. Moreover, every weak noose O in our setting will either separate an edge-less graph with three nodes from $V(\textsc{Sk}(b))$, or separate the graph $\textsc{Sk}(b, a)$ (where $a \in E(T_b)$) with at most one extra node from the rest of the graph.

Having defined weak nooses, we will now define our simplified operation. Let $A \oplus B$ be an exclusive or for two sets A and B, i.e. $A \oplus B = (A \cup B) \setminus (A \cap B)$. We will apply the $\oplus$-operation to weak nooses, whose $\oplus$ is again a weak noose. The following lemma, whose setting is illustrated in Figure 5, is central to our framework as it shows that we can always obtain the noose for the parent arc a_P from the nooses of the child arcs a_L and a_R using a short sequence of $\oplus$-operations such that every intermediate result is a weak noose. Therefore, using our framework it is now sufficient to show how to compute the set of types for a weak noose O from the set of types of two weak nooses O_1 and O_2 given that O can be obtained as $O_1 \oplus O_2$. This greatly simplifies the computation of types and can potentially be also applied to simplify dynamic programming algorithms on sphere-cut decompositions for other problems in the future.

Lemma 18. *Let a_P be a parent arc with two child arcs a_L and a_R in a sphere-cut decomposition $\langle T, \lambda, \Pi \rangle$ of a biconnected multi-graph G with the drawing D. There exists a sequence Q of at most 3 $\oplus$-operations such that:*

- *each step generates a weak noose O with $|O| \leq 1 + \max\{|mid(a_P)|, |mid(a_L)|, |mid(a_R)|\}$ as the $\oplus$-operation of two weak nooses O_1 and O_2, whose inside region contains all subcurves in $(O_1 \cap O_2)$,*
- *the last step generates the noose O_{a_P},*
- *Q contains O_{a_L} and O_{a_R} and at most two new weak nooses, each of them bounds the edge-less graph of size 3.*

Proof. Using Condition (ii) in the definition of sphere-cut decompositions together with certain planar properties, it is straightforward to show that, the intersection of any two of the nooses O_{a_P}, O_{a_L}, and O_{a_R}, interpreted as curves, is a single segment and that the intersection of all three contains at most two points. Therefore, we obtain the following fact:

(*) The difference between the curve O_{a_P} and the curve $O_{a_L} \cup O_{a_R}$ corresponds to at most two segments s_1 and s_2.

Let O' be the set of subcurves $O_{a_P} \oplus O_{a_L} \oplus O_{a_R}$. Note that O' is a union of simple closed curves (weak nooses) that do not share any subcurve, since it is obtained from $\oplus$-operations of simple closed curves and because O_{a_L} and O_{a_R} are inside O_{a_P} and the insides of O_{a_L} and O_{a_R} are disjoint; please also refer to Figure 5 for an illustration of the current setting. Since $\textsc{Sk}(b, a_P) = \textsc{Sk}(b, a_L) \cup \textsc{Sk}(b, a_R)$ and O' is outside of O_{a_L} and O_{a_R} but inside O_{a_P}, all weak nooses O from O' bound an edge-less graph. Therefore, all of the subcurves from O are inside the same face of D. This together with Condition (ii), implies that every weak noose O in O' contains exactly one subcurve from each O_{a_P}, O_{a_L} and O_{a_R}. Therefore, using the fact that the weak nooses in O' do not share any subcurve together with **(*)**, we obtain that O' contains at most two weak nooses (one for each of the at most two segments s_1 and s_2). In summary, O' is a union of at most two weak nooses and each of them bounds an edge-less graph of size 3.

Now we are going to define a sequence of operations Q that satisfies the conditions set out in the Lemma. In order to achieve this goal, we will make use of the following observation:

(**) If $|V(O_a) \cap V(O)| = 2$, then $O_a \oplus O$ is a weak noose, where O is a weak noose from O', and $a \in \{a_L, a_R\}$.

Below we distinguish between three cases, depending on the number of weak nooses O' is made of.

- If $O' = \emptyset$ then $Q = O_{a_L} \oplus O_{a_R}$.
- If O' consists of only one weak noose O_1, then we do the following. If $|V(O_{a_L}) \cap V(O_1)| = 2$, then we can use the sequence $Q = (O_{a_L} \oplus O_1) \oplus O_{a_R}$. Similarly, if $|V(O_{a_R}) \cap V(O_1)| = 2$, then we can use the sequence $Q = (O_{a_R} \oplus O_1) \oplus O_{a_L}$.
  Otherwise, $|V(O_{a_L}) \cap V(O_1)| = |V(O_{a_R}) \cap V(O_1)| = 3$. Let $c = (\{u, v\}, f)$ be the common subcurve of $O_{a_P} \cap O_1$. Then, $c \notin O_{a_L} \cup O_{a_R}$ and $u, v \in V(O_{a_L}) \cap V(O_{a_R})$.
  If $|O_{a_P}| = 1$, then G is not biconnected which is against our assumption, otherwise $|O_{a_P}| > 1$, but then $O_{a_P} \cap O_{a_L} \cap O_{a_R}$ is not empty, which contradicts our previous observation that the intersection of all three nooses O_{a_P}, O_{a_L}, and O_{a_R} (seen as curves) is at most two points.
- If O' consists of two weak nooses O_1 and O_2, then for every $a \in \{a_L, a_R\}$ and $i \in \{1, 2\}$, it holds that $O_a \oplus O_i$ is a weak noose, which can be seen as follows. Because of **(*)**, we obtain that $|V(O_a) \cap V(O_i)| = 2$ which together with **(**)** implies that $O_a \oplus O_i$ is a weak noose. Therefore, we can use the sequence $Q = (O_{a_L} \oplus O_1) \oplus (O_{a_R} \oplus O_2)$.

Note that, for each $\oplus$-operation between O_1 and O_2 in the solution Q, the region $O_1 \oplus O_2$ contains subcurves $O_1 \cap O_2$, because intersection between any two regions made from nooses from Q is only part of their boundaries. Moreover, to show that $|O| \leq 1 + \max\{|mid(a_P)|, |mid(a_L)|, |mid(a_R)|\}$ for every weak noose O obtained as an intermediate step, it suffices to consider case that O is obtained from one of the at most parts O'' of O' and O_a for $a \in \{a_L, a_R\}$. But this follows because $V(O'')$ contains only one vertex that is not in $V(O)$. ◻

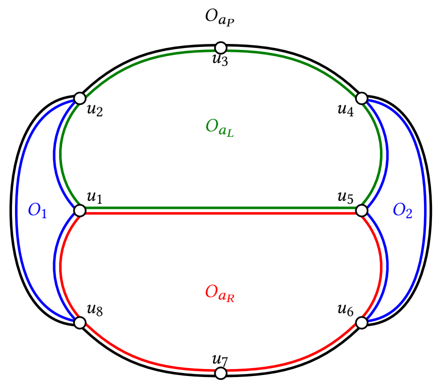

Fig. 5. An illustration of the relationship of the parent noose O_{a_P} and the child nooses O_{a_L} and O_{a_R}. The illustration represents the case of Lemma 18 where $O' = O_{a_P} \oplus O_{a_L} \oplus O_{a_R}$ consists of two disjoint weak nooses (triangles) O_1 and O_2. Let $c_{i,j}$ be a curve between u_i, u_j then all nooses are defined as follows: $O_{a_P} = \{c_{2,3}, c_{3,4}, c_{4,6}, c_{6,7}, c_{7,8}, c_{2,8}\}$, $O_{a_L} = \{c_{1,2}, c_{2,3}, c_{3,4}, c_{4,5}, c_{1,5}\}$, $O_{a_R} = \{c_{1,5}, c_{5,6}, c_{6,7}, c_{7,8}, c_{1,8}\}$, $O_1 = \{c_{1,2}, c_{2,8}, c_{1,8}\}$, $O_2 = \{c_{4,5}, c_{5,6}, c_{4,6}\}$, $O' = O_1 \cup O_2$.

We are now ready to define the types of weak nooses, which informally can be seen as a generalization of the types of nodes in an SPQR-tree introduced in Subsection 4.2. An illustration of the types is also provided in Figure 7. In the following we fix an arbitrary order π_G of the vertices in G. A type of a weak noose O is a triple (ψ, M, S) such that:

- ψ is a function that for each subcurve $c = (\{u, v\}, f)$ in O returns a sequence of at most two new nodes. Informally, these two nodes are on the subcurve c (in the order given by $\psi(c)$) and if $\pi_G(u) < \pi_G(v)$ we assume that $[u, \psi(c), v]$ is the sequence of nodes on the subcurve c.
- S is a subset of $V(O)$.
- $M \subseteq \{\, \{u, v\} \mid u, v \in V(\psi) \cup (V(O) \setminus S) \wedge u \neq v \,\}$, $V(\psi) \subseteq V(M)$, and M is a non-crossing matching w.r.t. the circular order $\pi^\circ(\psi)$ defined as follows. $\pi^\circ(\psi)$ is the circular order obtained from the circular order $\pi^\circ(O)$ of $V(O)$ after adding $\psi(c)$ between u and v, for every $c = (\{u, v\}, f) \in O$ assuming that $\pi_G(u) < \pi_G(v)$.

Let O be a weak noose and $X = (\psi, M, S)$ be a type. We define the graph $\text{Pe}_X(b, O)$ as the graph obtained from $\text{Pe}(b, O)$ after adding the vertices in $V(\psi)$ and all edges on the cycle $\pi^\circ(\psi)$. We say that O has type X if there is a set $\mathcal{P}$ of vertex-disjoint paths or a cycle in the complete graph with vertex set $V(\text{Pe}_X(b, O))$ such that:

- $\mathcal{P}$ consists of exactly one path P_e between u and v for every $e = \{u, v\} \in M$ or $\mathcal{P}$ is a cycle and $M = \emptyset$.
- $\{\, \text{IN}(P) \mid P \in \mathcal{P} \,\}$ is a partition of $\big(V(\text{Pe}(b, O)) \setminus V(O)\big) \cup S$, where $\text{IN}(P)$ denotes the set of inner vertices of P.
- there is a planar drawing D_X of $\text{Pe}_X(b, O) \cup \bigcup \mathcal{P}$ with outer-face f such that $\sigma(f) = \pi^\circ(\psi)$.

We also say that the fact that O has type X is *witnessed* by the pair $(\mathcal{P}, D_X)$. We say that type $X = (\psi, M, S)$ of O is the *full type*, if $M = \emptyset$ and $S = V(O)$, which informally means that $\mathcal{P}$ is a Hamiltonian cycle. Moreover we say that type X is the *empty type*, if $M = S = \emptyset$, which may only occur when $\text{Pe}(b, O)$ is merely an edge.

Lemma 19. *Let O be weak noose. Then, the number of types defined on O is at most $28^{|O|}$ and all possible types for O can be enumerated in time $O(28^{|O|}|O|)$.*

Proof. Let $X = (\psi, M, S)$ be a type that can be defined on a weak noose O. Let DW be the Dyck word corresponding to the matching M from Observation 11. For each $v \in V(O)$ there are 4 possibilities of the role of v in type X, i.e., $v \in S$, $v \notin S \cup V(M)$ or $v \in V(M)$ and v corresponds to either "[" or "]" in DW. Note that due to the type definition we get that $V(\psi) \subseteq V(M)$. Therefore for each $v \in V(\psi)$ there are 2 possibilities of the role of v in X, i.e., v corresponds to "[" or "]" in DW. For each subcurve $c \in O$, there are 3 possible values $\{\emptyset, [x], [x, x']\}$ for $\psi(c)$, and therefore there are $1 + 2 + 4 = 7$ possibilities, i.e., 1, 2, and 4 possibilities in case that $\psi(c) = \emptyset$, $\psi(c) = [x]$, and $\psi(c) = [x, x']$, respectively, of the role of c in type X. Furthermore, since $|O| = |V(O)|$, there are at most $4^{|O|}7^{|O|} = 28^{|O|}$ types that can be defined on O.

We can generate all types, by choosing a starting vertex on $V(O)$ together with a direction. We can then assign a role to each vertex in $V(O)$ and every subcurve of O and verify that the corresponding word is a Dyck word in time $O(|O|)$ and if so translate it into a type description using Observation 11. Since there are at most $28^{|O|}$ possibilities to check and each can be checked in time $O(|O|)$, we obtain $O(28^{|O|}|O|)$ as the total run-time to enumerate all possible types for O. □

Finally, we now defined the type of weak nooses for a given witness. Let $W = (D, D_H, G_H, H)$ be a witness for G that *respects* $\mathcal{T}$ and let b be an R-node or an S-node with sphere-cut decomposition $\langle T_b, \lambda_b, \Pi_b \rangle \in \mathcal{T}$. Then, the *type* of a weak noose $O \subseteq C(T_b)$ w.r.t. W, denoted by $\Gamma_W(b, O)$, is obtained as follows. Let D'_H be the drawing obtained from D_H after adding the noose O. Then, because W respects $\mathcal{T}$, it holds that every subcurve $c \in O$ is crossed at most twice in D'_H. In the following we will assume that we replaced every such crossing with a new vertex in D'_H and that these vertices are also introduced into G_H and H. Moreover, we let $\psi(c)$ be the sequence of (the at most two) new vertices introduced in this way for the subcurve $c = (\{u, v\}, f) \in O$ such that $[u, \psi(c), v]$ is the ordering of the vertices on c assuming that $\pi_G(u) < \pi_G(v)$. Let D^O_H be the drawing D'_H restricted to $\textsc{Pe}(b, O) \cup V(\psi)$ and let G^O_H and H^O be obtained in the same way from G_H and H, respectively. Let f_O be the face in D^O_H such that $V(f_O) = V(O) \cup V(\psi)$. Let $\mathcal{P}$ be a set of all maximal paths in H^O each of size at least 2. Then, S is the set of all vertices in $V(O)$ that have degree two in $\mathcal{P}$ and the matching M contains edge between the endpoints of every path in $\mathcal{P}$. Then, the type $\Gamma_W(b, O)$ is equal to the triple $X = (\psi, M, S)$. Note that X satisfies all properties of a type because of the following. First every node v in $V(\psi)$ has degree 1 in $\mathcal{P}$ and therefore $V(\psi) \subseteq V(M)$. Moreover, M is a non-crossing matching w.r.t. $\pi^\circ(\psi) = \sigma(f_O)$ because D^O_H is a planar drawing of $\mathcal{P}$ with face f_O. Therefore, the weak noose O has type X and this is witnessed by the pair $(\mathcal{P}, D^O_H)$.

5 An FPT-algorithm for SUBHAM using Treewidth

In this section we show that SUBHAM admits a constructive single-exponential fixed-parameter algorithm parameterized by treewidth.

Theorem 20. SUBHAM *can be solved in time* $2^{O(tw)} \cdot n^{O(1)}$*, where tw is the treewidth of the input graph.*

Since the treewidth of an n-vertex planar graph is upper-bounded by $O(\sqrt{n})$ [28, 43, 49] and there are single-exponential constant-factor approximation algorithms for treewidth [40], Theorem 20 immediately implies the following corollary.

Corollary 21. SUBHAM *can be solved in time* $2^{O(\sqrt{n})}$.

The main component used towards proving Theorem 20 is the following lemma.

Lemma 22. *Let G be a biconnected multi-graph with n vertices and m edges and SPQR-tree $\mathcal{B}$. Then, we can decide in time $O(315^\omega n + n^3)$ whether G is subhamiltonian, where ω is the maximum branchwidth of $\textsc{Sk}(b)$ over all R-nodes and S-nodes b of $\mathcal{B}$.*

With the help of Lemma 22, Theorem 20 can now be easily shown as follows.

Proof of Theorem 20. Let G be the graph given as input to SUBHAM having n vertices and m edges. Because of Theorem 17, we can assume that G is biconnected since we can otherwise solve every biconnected component of G independently. We first test whether G is planar, which is well-known do be achievable in linear-time [37]. If this is not the case, the algorithm correctly outputs no. Otherwise, the algorithm uses Lemma 5 to compute an SPQR-tree $\mathcal{B}$ of G with at most $O(m)$ nodes and edges inside skeletons in time at most $O(n+m)$; note that $O(m) = O(n)$ because G is planar. We then employ Lemma 22 to solve SUBHAM in time $O(315^{\omega} n + n^3)$, where ω is the maximum branchwidth of $\text{Sk}(b)$ over all R-nodes and S-nodes b of $\mathcal{B}$. Since ω is an upper bound on the branchwidth of G, we obtain from Lemma 7 that the branchwidth of G is at most the treewidth $tw(G)$ of G plus 1, which implies that SUBHAM can be solved in time $O(315^{tw(G)} n + n^3)$, as required. □

The remainder of this section is therefore devoted to a proof of Lemma 22, which we show by providing a bottom-up dynamic programming algorithm along the SPQR-tree of the graph. That is, let G be a biconnected multi-graph, $\mathcal{B}$ be an SPQR-tree of G with associated set $\mathcal{T}$ of sphere-cut decompositions for every R-node and S-node of $\mathcal{B}$. Using a dynamic programming algorithm starting at the leaves of $\mathcal{B}$, we will compute a set $\mathcal{R}(b)$ of all types X satisfying the following two conditions:

(R1) If $X \in \mathcal{R}(b)$, then b has type X.

(R2) If there is a witness $W = (D, D_H, G_H, H)$ for G that *respects* $\mathcal{B}$ such that b has type $X = \Gamma_W(b)$, then $X \in \mathcal{R}(b)$.

Interestingly, we do not know whether it is possible to compute the set of all types X such that b has type X as one would usually expect to be able to do when looking at similar algorithms based on dynamic programming. That is, we do not know whether one can compute the set of types that also satisfies the reverse direction of *(R1)*. While we do not know, we suspect that this is not the case because b might have a type that can only be achieved by crossing some sub-curves of nooses inside of $\text{Pe}(b)$ more than twice. Indeed Lemma 15, which allows us to avoid more than two crossings per sub-curve, requires the property that the type of b can be extended to a Hamiltonian cycle of the whole graph, which is clearly not necessarily the case for every possible type of b.

This section is organized as follows. First in Subsection 5.1, we show how to compute $\mathcal{R}(b)$ for every P-node b of $\mathcal{B}$. This is probably the most challenging part of the algorithm and we show that instead of having to enumerate all possible orderings among the children of b in $\mathcal{B}$, we merely have to consider a constant number children and their orderings. This allows us to compute $\mathcal{R}(b)$ very efficiently in time $O(\ell)$, where ℓ is the number of children of b in $\mathcal{B}$. Then, in Subsection 5.2, we show how to compute $\mathcal{R}(b)$ for any R-node and S-node b of $\mathcal{B}$ using a dynamic programming algorithm on a sphere-cut decomposition of $\text{Sk}(b)$. We then put everything together and show Lemma 22 in Subsection 5.3.

5.1 Handling P-nodes

In this part, we show how to compute the set of types for any P-node in the given SPQR-tree by establishing the following lemma.

Lemma 23. *Let b be a P-node of $\mathcal{B}$ such that $\mathcal{R}(c)$ has already been computed for every child c of b in $\mathcal{B}$. Then, we can compute $\mathcal{R}(b)$ in time $O(\ell)$, where ℓ is the number of children of b in $\mathcal{B}$.*

In the following, let b be a P-node of $\mathcal{B}$ with reference edge (s, t) and let C with $|C| = \ell$ be the set of all children of b in $\mathcal{B}$. Informally, $\mathcal{R}(b)$ is the set of types X such that there is an ordering $\rho = (c_1, \ldots, c_\ell)$ of the children in C and

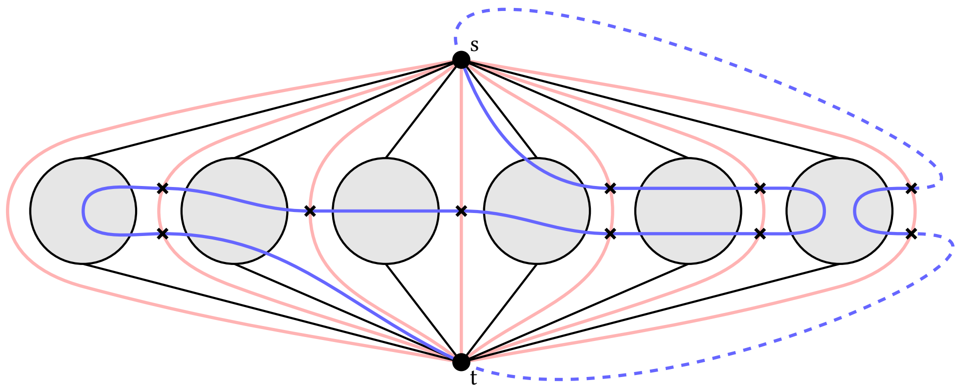

Fig. 6. An illustration of how a Hamiltonian Cycle in normal form can interact with a drawing of $\text{Pe}(b)$ for a P-node b of $\mathcal{B}$. Here, the pertinent graphs $\text{Pe}(c)$ for all children c of b (without the nodes s and t of the common reference edge (s,t)) are represented by gray ellipses. The Hamiltonian cycle is given in blue with dashed segments representing path segments outside of $\text{Pe}(b)$. The red curves represent the subcurves of N_c for every child c of b. Note again that the subcurves R_c and $L_{c'}$ are identical for every two children c and c' such that $\text{Pe}(c)$ is drawn immediately to the left of $\text{Pe}(c')$. The drawing is in normal form, *i.e.*, is a drawing that respects the nooses N_c for every child c of b, because every red curve is intersected by H at most twice. In this figure all but the types of the second and fourth pertinent graph are clean. Moreover, the type of the third and fifth pertinent graphs are 1-good and 2-good, respectively, and the types of all other pertinent graphs are bad.

an assignment $\tau : C \to \mathcal{X}$ of children to types with $\tau(c) \in \mathcal{R}(c)$ for every child $c \in C$ that "realizes" the type X for b. The main challenge is to compute $\mathcal{R}(b)$ efficiently, i.e., without having to enumerate all possible orderings ρ and assignments τ. Below, we make this intuition more precise before proceeding.

For a type $X = (\psi, M, S)$ of b and $A \in \{L, R\}$, we let $\#_A(X) = |\psi(A)|$. Moreover, for every $A \in \{s,t\}$, we set $\#_A(X)$ to be equal to 2 if $A \in S$, equal to 1 if $A \in V(M)$ and equal to 0 otherwise. Next, let $\rho = (X_1, \dots, X_\ell)$ be a sequence of types, where $X_i = (\psi_i, M_i, S_i)$ for every i with $1 \le i \le \ell$. We say that ρ is *weakly compatible* if the following holds:

(C1) for every i with $1 \le i < \ell$, $\#_R(X_i) = \#_L(X_{i+1})$, and
(C2) $\sum_{i=1}^{\ell} \#_s(X_i) \le 2$ and $\sum_{i=1}^{\ell} \#_t(X_i) \le 2$.

Note that *(C1)* corresponds to our assumption made in Lemma 10 that we can add the nooses N_b to any planar drawing D of G such that every face of D contains at most one subcurve of any N_b. This in particular means that if $\text{Pe}(c)$ is drawn immediately to the left of $\text{Pe}(c')$ for two children c and c' of b, then the subcurves R_c and $L_{c'}$ are identical. Please also refer to Figure 6 for an illustration of these subcurves.

Let ρ be weakly compatible. We define the following auxiliary graph $H(\rho)$. $H(\rho)$ has two vertices s and t and additionally for every i with $1 \le i \le \ell$ and every vertex $v \in V(\psi)$, $H(\rho)$ has a vertex v_i. For convenience, we also use s_i and t_i to refer to s and t, respectively. Moreover, $H(\rho)$ has the following edges:

- for every $1 \le i \le \ell$ if $M_i = \emptyset$ and $S_i = \{s_i, t_i\}$, $H(\rho)$ has a cycle on s_i and t_i,
- for every $1 \le i \le \ell$ if $M_i \ne \emptyset$ then for every $e = \{u, v\} \in M_i$, $H(\rho)$ has the edge $\{u_i v_i\}$,

- for every $1 \leq i < \ell$, $H(\rho)$ contains the edge $\{r_i, l_{i+1}\}$ if $r \in \psi_i(R)$ and $l \in \psi_{i+1}(L)$,
- for every $1 \leq i < \ell$, $H(\rho)$ contains the edge $\{r'_i, l'_{i+1}\}$ if $r' \in \psi_i(R)$ and $l' \in \psi_{i+1}(L)$.

Lemma 24. *Let ρ be weakly compatible. Then, $H(\rho)$ is planar.*

Proof. Because M_i is a non-crossing matching w.r.t. the cyclic ordering (s, r, r', t, l', l) for every $i \in [1, \ell]$, it holds that the graph $H(\rho)$ induced by the vertices in $\{s_i, t_i\} \cup V(\psi_i)$ has a planar drawing D_i, where $\psi_i(L)$ are placed to the east, s is placed in the north, $\psi_i(R)$ is placed in the west, and t is placed on the south. Taking the disjoint union of the drawings D_i ordered $D_1, \ldots, D_\ell$ from east to west and identifying all $s'_i s$ with s and all t_i's with t, then gives a planar drawing of $H(\rho)$. □

We say that ρ is *compatible* if it is weakly compatible and furthermore either $H(\rho)$ is acyclic, or $H(\rho) - (\bigcup_{i=1}^{\ell} S_i)$ is a single (Hamiltonian) cycle.

Lemma 25. *Let ρ be compatible such that $H(\rho)$ is acyclic. Then, $H(\rho)$ is the disjoint union of paths whose endpoints are in $\{s, t, l_1, l'_1, r_\ell, r'_\ell\}$. Moreover, no vertex in $\{l_1, l'_1, r_\ell, r'_\ell\}$ can be an inner vertex of those paths.*

Proof. We first show that the degree of every vertex in $H(\rho)$ is at most two. Because of *(C2)* this clearly holds for the vertices s and t. Moreover, every vertex in $v \in \{l_i, l'_i, r_i, r'_i\}$ for any i with $1 \leq i \leq \ell$ has exactly one neighbor among $\{l_i, l'_i, r_i, r'_i, s, t\}$ and at most one neighbor in $V(H(\rho)) \setminus \{l_i, l'_i, r_i, r'_i, s, t\}$. Therefore, $H(\rho)$ has maximum degree at most two and since $H(\rho)$ is acyclic, $H(\rho)$ is a disjoint union of paths. Moreover, the vertices $\{l_1, l'_1, r_\ell, r'_\ell\}$ have degree exactly one and hence cannot be inner vertices of the paths. Finally, since every vertex of $H(\rho)$ apart from the vertices $\{s, t, l_1, l'_1, r_\ell, r'_\ell\}$ must have degree exactly two, only these vertices can act as endpoints of the paths. □

In the following let $\rho = (X_1, \ldots, X_\ell)$ be compatible. We now define the type X associated with ρ, which we denote by $X(\rho)$, as follows. If $H(\rho)$ is a single cycle and $\{s, t\} \subseteq \bigcup_{i=1}^{\ell} S_i$, then we set $X(\rho) = (\psi, \emptyset, \{s, t\})$, where $\psi(L) = \psi(R) = \emptyset$. Otherwise, let $\mathcal{P}(\rho)$ be the set of paths in $H(\rho)$, which due to Lemma 25 have their endpoints in $\{s, t, l_1, l'_1, r_\ell, r'_\ell\}$. Then, we set $X(\rho) = (\psi, M, S)$, where ψ, M, and S are defined as follows. M contains the set $\{u, v\}$ for every path in $\mathcal{P}(\rho)$ with endpoints u and v; for brevity, we denote $l_1, l'_1, r_\ell, r'_\ell$ as l, l', r, r', respectively. Moreover, $\psi(L) = V(M) \cap \{l, l'\}$, $\psi(R) = V(M) \cap \{r, r'\}$, and S contains s (t) if $\sum_{i=1}^{\ell} \#_s(X_i) = 2$ ($\sum_{i=1}^{\ell} \#_t(X_i) = 2$). This completes the definition of $X(\rho)$, which can be easily seen to be a type for b because $G(\rho)$ is planar due to Lemma 24.

We say that ρ is *realizable* if there is an ordering $\pi = (c_1, \ldots, c_\ell)$ of the children in C and an assignment $\tau : C \to \mathcal{X}$ from children to types with $\tau(c) \in \mathcal{R}(c)$ for every $c \in C$ such that $\rho = \tau(\pi) = (\tau(c_1), \ldots, \tau(c_\ell))$. Below, we prove that if ρ is a compatible and realizable, then b has type $X(\rho)$.

Lemma 26. *Let ρ be compatible and realizable. Then, b has type $X(\rho)$.*

Proof. Let (π, τ) with $\pi = (c_1, \ldots, c_\ell)$ be the ordering and assignment that witnesses that ρ is realizable. Since $\tau(c) = (\psi_c, M_c, S_c) \in \mathcal{R}(c)$ and $\mathcal{R}(c) \subseteq \mathcal{X}(c)$ (using *(R1)* in the definition of $\mathcal{R}(c)$) for every child c of b, we obtain that there is a set $\mathcal{P}_c$ of vertex-disjoint paths in the complete graph with vertex set $V(\text{Pe}^*(c))$ such that:

- $\mathcal{P}_c$ consists of exactly one path P_e between u and v for every $e = \{u, v\} \in M_c$.
- $\{\, \text{IN}(P) \mid P \in \mathcal{P}_c \,\}$ is a partition of $(V(\text{Pe}(b)) \setminus \{s, t\}) \cup S_c$, where $\text{IN}(P)$ denotes the set of inner vertices of the path P.
- there is a planar drawing $D(b, \tau(c))$ of $\text{Pe}^*(b) \cup \bigcup_{P \in \mathcal{P}_c} P$ with outer-face f such that $\sigma(f) = \{s, r, r', t, l', l\}$.

Moreover, since $\sigma(f) = \{s, r, r', t, l', l\}$, we can (and will) in the following assume that the drawing $D(c, \tau(c))$ has: s at its north, t at its south, l and l' at its west with l being north of l' and r and r' at its east with r to the north of r'.

Let D be the planar drawing obtained from the disjoint union of the drawings $D(c_1, \tau(c_1)), \ldots, D(c_\ell, \tau(c_\ell))$ by drawing them in the order given by ρ from west to east without overlap. To avoid name clashes between vertices, we refer to the vertex $v \in \{s, t, l, l', r, r'\}$ belonging to the drawing $D(c_i, \tau(c_i))$ inside D as v_i.

Because of the above mentioned properties of the drawings $D(c, \tau(c))$, we can now add (and draw) the following edges to D without crossings to obtain the planar drawing D'.

- The edges of the paths $(s_1, \ldots, s_\ell)$ and $(t_1, \ldots, t_\ell)$,
- For every i with $1 \leq i < \ell$, the edges $\{r_i, l_{i+1}\}$ and $\{r'_i, l'_{i+1}\}$.

Let D'' be the planar drawing obtained from D' after:

- contracting the path $(s_1, \ldots, s_\ell)$ into the fresh vertex s,
- contracting the path $(t_1, \ldots, t_\ell)$ into the fresh vertex t,
- For every i with $1 \leq i \leq \ell$:
  - if $i \neq \ell$ and $|\psi_i(R)| = 1$ contract the edge $\{r_i, r'_i\}$ into the vertex r_i,
  - if $i \neq \ell$ and $|\psi_i(R)| = 0$ remove the vertices r_i and r'_i,
  - if $i \neq 1$ and $|\psi_i(L)| = 1$ contract the edge $\{l_i, l'_i\}$ into the vertex l_i,
  - if $i \neq 1$ and $|\psi_i(L)| = 0$ remove the vertices l_i and l'_i.
- removing all edges of the form sl_i, $l_il'_i$, l'_it, sr_i, $r_ir'_i$, and r'_it for every $i \notin \{1, \ell\}$.

Let D_b be the planar drawing obtained from D'' after:

- contracting all edges incident to any vertex in $\{l_i, l'_i \mid 1 < i \leq \ell\} \cup \{r_i, r'_i \mid 1 \leq i < \ell\}$,
- renaming the vertices l_1, l'_1, r_ℓ, and r'_ℓ to l, l', r, and r'.

Let H_b be the planar graph corresponding to D_b.

Let D_ρ be the planar drawing obtained from D'' after:

- contracting every path $P \in \mathcal{P}_c$ for every $c \in C$ into a single edge,
- if $|\psi_1(L)| = 1$ contract the edge $\{l_1, l'_1\}$ into the vertex l_1,
- if $|\psi_1(L)| = 0$ remove the vertices l_1 and l'_1,
- if $|\psi_\ell(R)| = 1$ contract the edge $\{r_\ell, r'_\ell\}$ into the vertex r_ℓ,
- if $|\psi_\ell(R)| = 0$ remove the vertices r_ℓ and r'_ℓ,

Let H_ρ be the planar graph corresponding to D_ρ.

Then, H_ρ is isomorphic to $G(\rho)$ and D_b is a planar drawing of $\textsc{Pe}^*(b) \cup \bigcup_{P \in \bigcup_{c \in C} \mathcal{P}_c}$ that witnesses that b has type $X(\rho)$. □

Lemma 27. *Let $W = (D, D_H, G_H, H)$ be a witness for G that* respects *$\mathcal{B}$ and $\mathcal{T}$. Then, there is a realizable and compatible ρ such that $X(\rho) = \Gamma_W(b)$.*

Proof. Let $\pi = (c_1, \ldots, c_\ell)$ be the ordering of the children in C according to the drawing D_H. Moreover, let $\tau : C \to \mathcal{X}$ be the assignment defined by setting $\tau(c) = \Gamma_W(c)$ for every $c \in C$. Note that because of *(R2)* in the definition of $\mathcal{R}(c)$ it holds that $\tau(c) \in \mathcal{R}(c)$ for every $c \in C$. Finally, let ρ be the sequence $(\tau(c_1), \ldots, \tau(c_\ell))$. Then, ρ is clearly realizable. Moreover, ρ is compatible since W respects $\mathcal{B}$ and since H is a Hamiltonian cycle. □

From Lemmas 26 and 27, we now obtain the following corollary.

Corollary 28. *The set R containing every type $X \in \mathcal{X}$ such that there is a compatible and realizable ρ with $X = X(\rho)$ satisfies the properties* (R1) *and* (R2).

Therefore, from now onward we can focus on finding the set of all types X for which there is a compatible and realizable ρ such that $X = X(\rho)$.We will now show that this can be achieved very efficiently because only a constant number, i.e., at most 8 types (and their ordering) need to be specified in order to infer the type of a sequence ρ. Let $X = (\psi, M, S) \in \mathcal{X}$ be a type. We say that X is *dirty* if $\#_s(X) + \#_t(X) > 0$ and otherwise we say that X is *clean*. We say that X is 0-*good*, 1-*good*, and 2-*good*, if X is clean and additionally $M = \emptyset$, $M = \{\{l, r\}\}$, and $M = \{\{l, r\}, \{l', r'\}\}$, respectively. We say that X is *good* if it is x-good for some $x \in \{0, 1, 2\}$ and otherwise we say that X is *bad*. We denote by $\mathcal{X}_G$ and $\mathcal{X}_B$ the subset of $\mathcal{X}$ consisting only of the good respectively bad types. An illustration of these notions is provided in Figure 6.

Lemma 29. *Let $\rho = (X_1, \ldots, X_\ell)$ be compatible. Then, ρ contains at most 4 dirty types and at most 4 types that are clean and bad.*

Proof. Let $\rho = (X_1, \ldots, X_\ell)$ with $X_i = (\psi_i, M_i, S_i)$ be compatible. The statement that ρ contains at most 4 dirty types follows directly from *(C2)* in the definition of weak compatibility. It remains to show that ρ contains at most 4 types that are clean and bad. First note that if type $X = (\psi, M, S)$ is clean and bad, then either $M = \{\{L, L\}\}$, $M = \{\{R, R\}\}$, or $M = \{\{L, L\}, \{R, R\}\}$. Now suppose for a contradiction that ρ contains at least 5 types that are clean and bad. Then, there are indices $1 \leq i < j < k \leq \ell$ such that M_i, M_j, and M_k either all contain the pair $\{L, L\}$ or all of them contain the pair $\{R, R\}$. Let us assume the former case since the argument for the latter case is analogous. Because ρ is compatible the path/cycle P in $H(\rho)$ that contains the edge $\{l_j, l'_j\}$ must also contain s and t. This is because if P does not contain s and t it can only go to the left until it is blocked by the path containing the edge $\{l_i, l'_i\}$. The same holds for the path/cycle in $H(\rho)$ that contains the edge $\{l_k, l'_k\}$. Therefore, $\{l_j, l'_j\}$ and $\{l_k, l'_k\}$ are contained together with s and t on a cycle C in $H(\rho)$. Finally, because of *(C2)* in the definition of weakly compatible, we obtain that C does not contain the edge $\{l_i, l'_i\}$, which contradicts our assumption that $H(\rho)$ is either a single cycle or acyclic. ◻

The following corollary follows immediately from Lemma 29 since every bad type is either clean or dirty.

Corollary 30. *Let $\rho = (X_1, \ldots, X_\ell)$ be compatible, then ρ contains at most 8 bad types.*

Moreover, since deciding whether ρ is compatible merely requires us to check that ρ satisfies *(C1)* and *(C2)* and that either $H(\rho)$ is acyclic or $H(\rho) - (\bigcup_{i=1}^{\ell} S_i)$ is a single (Hamiltonian) cycle, we observe:

Observation 31. *It is possible to decide whether a given $\rho = (X_1, \ldots, X_\ell)$ is compatible in time $O(\ell)$.*

Next, we will show that any compatible sequence contains at most 8 bad types and that the type $X(\rho)$ is already determined by looking only at the sequence of bad types that occur in ρ. This will then allow us to simulate the enumeration of all possible sequences, by enumerating merely all sequences of at most 8 bad types.

We say that a sequence ρ' is an extension of ρ if ρ is a (not necessarily consecutive) sub-sequence of ρ'. We call a compatible sequence ρ (X, i)-*extendable* for some $X \in \mathcal{X}$ and integer i, if there is a compatible extension ρ' of ρ such that ρ' is obtained by adding i elements of type X to ρ and $X(\rho) = X(\rho')$. We call ρ X-*extendable* if ρ is (X, i)-extendable for any integer i. We say that ρ' is an (X, i)-*extension* of ρ if ρ' is a compatible sequence obtained after adding i elements of type X to ρ and $X(\rho) = X(\rho')$.

Lemma 32. *Let $\rho = (X_1, \dots, X_\ell)$ with $X_i = (\psi_i, M_i, S_i)$ and $X \in \mathcal{X}_G$. Then, ρ is $(X, 1)$-extendable if and only if ρ is X-extendable. Moreover, deciding whether ρ is $(X, 1)$-extendable and if so computing an (X, i)-extension ρ' of ρ can be achieved in time $O(\ell + i)$ for every integer i.*

Proof. The first statement of the lemma follows because if ρ' is a compatible extension of ρ containing at least one x-good type X_i, then we can add another x-good type immediately after or before X_i without violating the compatibility. Moreover, we can also delete X_i from ρ' without violating its compatibility.

Moreover, it is straightforward to verify that ρ can be extended by 1 x-good type if and only if either: (1) $|\psi_1(L)| = x$, (2) $|\psi_\ell(R)| = x$, or there is an index i with $1 \leq i < \ell$ such that $|\psi_i(L)| = |\psi_{i+1}(R)| = x$. This can clearly be tested in time $O(\ell)$ and if the test succeeds, it is also easy to compute an (X, i)-extension by adding all i elements of type X in one of the possible positions. ◻

Lemma 33. *Let ρ be a compatible sequence and let ρ' be the sub-sequence of ρ consisting only of the bad types in ρ. Then, ρ' is compatible and $X(\rho) = X(\rho')$.*

Proof. The lemma holds because removing any good type X preserves compatibility and does not change the type of the sequence; this is because neither s nor t are used by X and moreover $\#_L(X) = \#_R(X)$. ◻

At this point, we are ready to describe the algorithm we will use to compute $\mathcal{R}(b)$ (and argue its correctness). The algorithm first enumerates all possible compatible sequences ρ of at most 8 bad types, i.e., $\rho = (Y_1, \dots, Y_r)$ with $r \leq 8$ and $Y_i \in \mathcal{X}_B$ for every i. Note that there are at most $(|\mathcal{X}_B| + 1)^8$ (and therefore constantly many) such sequences and those can be enumerated in constant time. Given one such sequence $\rho = (Y_1, \dots, Y_r)$, the algorithm then tests whether the sequence can be realized given the types available for the children in C as follows. It first uses Lemma 32 to test whether ρ allows for adding a 0-good, 1-good or 2-good type in constant time. Let $A_\rho \subseteq \mathcal{X}_G$ be the set of all good types that can be added to ρ and let C_ρ be the subset of C containing all children c such that $A_\rho \cap \mathcal{R}(c) \neq \emptyset$.

Consider the following bipartite graph Q_ρ having one vertex y_i for every i with $1 \leq i \leq r$ representing the type Y_i on one side and one vertex v_c for every $c \in C$ representing the child c on the other side of the bipartition. Moreover, Q_ρ has an edge between y_i and v_c if $Y_i \in \mathcal{R}(c)$. We claim that ρ can be extended to a compatible and realizable sequence if and only if Q_ρ has a matching that saturates $\{y_1, \dots, y_r\} \cup \{ v_c \mid c \in C \setminus C_\rho \}$. This problem can be solved using a simple reduction to the well-known maximum flow problemas shown by the following lemma.

Lemma 34. *Let Q be a bipartite graph with partition $\{A, B\}$ and let $V \subseteq B$. There is an algorithm that in time $O(|E(Q)||A|)$ decides whether Q has a matching that saturates $A \cup V$.*

Proof. We solve the problem using a reduction to the maximum flow problem, which can be solved in time $O(mU)$ for a flow network with m edges with every edge having integer capacity at most U [23]. Let N be the network obtained as follows. The vertices of N are new vertices s, t, t' plus the vertices of Q. Moreover, N contains the following arcs:

- an arc from s to a with capacity 1 for every $a \in A$,
- an arc from a to b with capacity 1 for every edge $\{a, b\} \in E(Q)$,
- an arc from v to t with capacity 1 for every $v \in V$,
- an arc from t' to t with capacity $|A| - |V|$,
- an arc from b to t' with capacity 1 for every $b \in B \setminus V$.

It is now straightforward to show that Q has a matching that saturates $A \cup V$ if and only if N has an integer flow from s to t with value $|A|$. Since $U \leq |A|$ and $m = O(|V(Q)| + |E(Q)|)$, we obtain that our problem can be decided in time $O(|E(Q)||A|)$, which shows the stated run-time. ◻

The following lemma now establishes the correctness (i.e., the soundness and completeness) of the algorithm.

Lemma 35. *Let $X \in \mathcal{X}$. Then, there is a compatible and realizable sequence ρ with $X = X(\rho)$ if and only if there is a compatible sequence $\rho = (Y_1, \ldots, Y_r)$ of bad types with $r \leq 8$ with $X = X(\rho)$ such that the bipartite graph H_ρ has a matching that saturates $\{y_1, \ldots, y_r\} \cup \{\, v_c \mid c \in C \setminus C_\rho \,\}$.*

Proof. Towards showing the forward direction, let ρ be a compatible and realizable sequence and let $\rho' = (Y_1, \ldots, Y_r)$ be the sub-sequence of ρ containing only the bad types in ρ. Because of Corollary 30, it holds that $r \leq 8$ and because of Lemma 33, we have that ρ' is compatible. It remains to show that H_ρ has a matching that saturates $\{y_1, \ldots, y_r\} \cup (C \setminus C_\rho)$. Let (π, τ) be the ordering and assignment that witnesses that ρ is realizable. Let $\pi' = (c_1, \ldots, c_r)$ be the subsequence of π containing only the children c such that $\tau(c)$ is bad; note that $\rho' = \tau(\pi) = (\tau(c_1), \ldots, \tau(c_r))$. Then, $M = \{\{y_i, v_{c_i}\} \mid 1 \leq i \leq r\}$ is a matching in H_ρ that saturates $\{y_1, \ldots, y_r\} \cup (C \setminus C_\rho)$.

Towards showing the reverse direction, let $\rho' = (Y_1, \ldots, Y_r)$ be a compatible sequence of bad types with $r \leq 8$ and let M be the matching in H_ρ that saturates $\{y_1, \ldots, y_r\} \cup (C \setminus C_\rho)$. For convenience, we represent M as the bijective function $\tau' : C' \to \{Y_1, \ldots, Y_r\}$, where $C' = V(M) \cap C$ and such that $M = \{\{c, \tau'(c)\} \mid c \in C'\}$. Let $\tau : C \to \mathcal{X}$ be the assignment of children to types given by $\tau(c) = \tau'(c)$ for every $c \in C'$ and $\tau(c) \in (A_\rho \cap \mathcal{R}(c))$ for every $c \in C \setminus C'$. Because $C \setminus C' \subseteq C_\rho$, it holds that $A_\rho \cap \mathcal{R}(c) \neq \emptyset$ for every $c \in C \setminus C'$ and therefore it is possible to assign $\tau(c)$. We can now use Lemma 32 to obtain a compatible extension ρ of ρ' with $X(\rho') = X(\rho)$ such that ρ is obtained from ρ' after adding $|\tau^{-1}(X)|$ elements of type X for every $X \in A_{\rho'}$. Due to the choice of ρ and τ, there now exists an ordering π of C such that $\rho = \tau(\pi)$, which together with τ shows that ρ is also realizable. ◻

We are now ready to prove the central lemma of this section.

Proof of Lemma 23. We need to show that given $\mathcal{R}(c)$ for every child $c \in C$, we can compute $\mathcal{R}(b)$ in time $O(\ell)$.

Because of Corollary 28 we can compute $\mathcal{R}(b)$ by computing all types $X \in \mathcal{X}$ such that there is a compatible and realizable sequence ρ of types with $X = X(\rho)$. Moreover, because of Lemma 35, a type $X \in \mathcal{X}$ has such a compatible and realizable sequence ρ if and only if there is a compatible sequence $\rho = (Y_1, \ldots, Y_r)$ of bad types with $r \leq 8$ and $X = X(\rho)$ such that the bipartite graph H_ρ has a matching that saturates $\{y_1, \ldots, y_r\} \cup \{\, v_c \mid c \in C \setminus C_\rho \,\}$ and this can be achieved by the following algorithm.

The algorithm first enumerates all possible compatible sequences ρ of at most 8 bad types, i.e., $\rho = (Y_1, \ldots, Y_r)$ with $r \leq 8$ and $Y_i \in \mathcal{X}_B$ for every i. Note that there are at most $(|\mathcal{X}_B| + 1)^8$ (and therefore constantly many) sequences of at most 8 bad types and because of Observation 31 checking whether such a sequence is compatible can be achieved in constant time. Therefore, all such compatible sequences can be enumerated in constant time. Given one such sequence $\rho = (Y_1, \ldots, Y_r)$, the algorithm first uses Lemma 32 to compute the set $A_\rho \subseteq \mathcal{X}_G$ (i.e., the set of all good types that can be added to ρ) in constant time. It then computes C_ρ (i.e., the subset of C containing all children c such that $A_\rho \cap \mathcal{R}(c) \neq \emptyset$) and constructs the bipartite graph H_ℓ in time $O(\ell)$. Finally, it uses Lemma 34 to decide whether H_ρ has a matching that saturates $\{y_1, \ldots, y_r\} \cup \{\, v_c \mid c \in C \setminus C_\rho \,\}$. If so, the algorithm correctly adds the type $X(\rho)$ to $\mathcal{R}(b)$ and otherwise the algorithm continues with the next sequence $\rho = (Y_1, \ldots, Y_r)$ with $r \leq 8$ and $Y_i \in \mathcal{X}_B$.

As pointed out above, the correctness of the algorithm follows from Corollary 28 and lemmas 32 and 35. The total run-time of the algorithm is dominated by the time required to decide whether H_ρ has a matching saturating

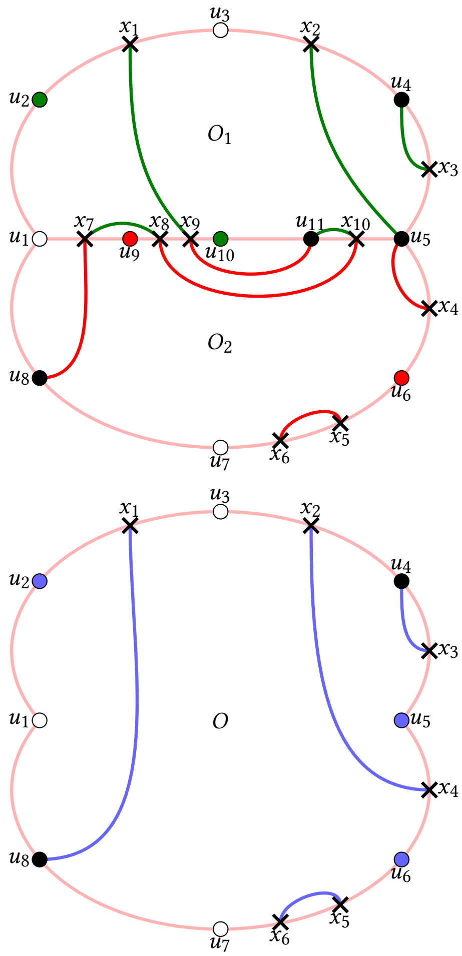

Fig. 7. **(Left)**An illustration of combining two compatible types $X_1 = (\psi_1, M_1, S_1)$ and $X_2 = (\psi_2, M_2, S_2)$ for two weak nooses O_1 and O_2 into the combined type $X = (\psi, M, S) = X_1 \circ X_2$ for $O = O_1 \oplus O_2$. Vertices of the graph are represented as circles and vertices subdividing the nooses, i.e., vertices in $V(\psi_1) \cup V(\psi_2)$, are represented as crosses. Black vertices are the vertices that are within a matching, i.e., the vertices in $V(M_1) \cup V(M_2)$, green (red) vertices are the vertices in S_1 (S_2) and all other vertices of the graph are white. The following holds for the type X_1 and X_2: $V(\psi_1) = \{x_1, x_2, x_3, x_7, x_8, x_9, x_{10}\}$, $V(\psi_2) = \{x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$, $M_1 = \{\{x_1, x_9\}, \{x_2, u_5\}, \{u_4, x_3\}, \{x_{10}, u_{11}\}, \{x_8, x_7\}\}$, $M_2 = \{\{x_7, u_8\}, \{x_8, x_{10}\}, \{x_9, u_{11}\}, \{u_5, x_4\}, \{x_5, x_6\}\}$, $S_1 = \{u_2, u_{10}\}$, and $S_2 = \{u_9, u_6\}$. **(Right)** The resulting type X of O for which the following holds: $V(\psi) = \{x_1, \dots, x_6\}$, $M = \{\{x_1, u_8\}, \{x_2, x_4\}, \{u_4, x_3\}, \{x_5, x_6\}\}$, and $S = \{u_2, u_5, u_6\}$.

$\{y_1, \dots, y_r\} \cup \{\, v_c \mid c \in C \setminus C_\rho \,\}$, which because of Lemma 34 can be achieved in time $O(|E(H_\rho)|r)$. Since $|E(H_\rho)| \leq 8\ell$ and $r \leq 8$ this term is equal to $O(\ell)$, as claimed. □

5.2 Handling R-nodes and S-nodes

Here, we will show how to compute the set of types satisfying *(R1)* and *(R2)* for every R-node and S-node of $\mathcal{B}$. To achieve this we will again use a dynamic programming algorithm albeit on a sphere-cut decomposition of $\text{Sk}(b)$ instead of on the SPQR-tree. The aim of this subsection is therefore to show the following lemma.

Lemma 36. *Let b be an R-node or S-node of $\mathcal{B}$ such that $\mathcal{R}(c)$ have already been computed for every child c of b in $\mathcal{B}$. Then, we can compute $\mathcal{R}(b)$ in time $O(315^{\omega}\ell + \ell^3)$, where ω is the branchwidth of the graph $\text{Sk}(b)$ and ℓ is the number of children of b in $\mathcal{B}$.*

In the following, let b be an R-node or S-node of $\mathcal{B}$ with reference edge (s_b, t_b) and let $\langle T_b, \lambda_b, \Pi_b \rangle$ be a sphere-cut decomposition of $\text{Sk}(b)$ that is rooted in $r = \lambda^{-1}((s_b, t_b))$. For a weak noose $O \subseteq C(T_b)$, let $\mathcal{A}(O)$ be the set of all types of O satisfying the following two natural analogs of *(R1)* and (R2), i.e.:

(RO1) If $X \in \mathcal{A}(O)$, then O has type X.

(RO2) If there is a witness (D, D_H, G_H, H) for G that *respects* $\mathcal{B}$ such that $\Gamma_W(b, O) = X$, then $X \in \mathcal{A}(O)$.

Our aim is to compute $\mathcal{A}(O_{a^r})$ for the arc a^r incident to the root r of T_b. We will achieve this by computing $\mathcal{A}(O_a)$ for every arc a of T_b via a bottom-up dynamic programming algorithm along T_b. Note that there is a one-to-one correspondence between the arcs of T_b that are connected to a leaf and the children of b in $\mathcal{B}$, i.e., the arc of T_b incident to leaf l corresponds to the child c of b representing the edge $\lambda(l)$. We start with two simple lemmas showing that: (1) We can compute $\mathcal{A}(O_a)$ for every arc of T_b incident to a leaf l of T_b in linear-time from $\mathcal{R}(c)$, where c is the child of b in $\mathcal{B}$ corresponding the edge $\lambda(l)$ and (2). We can compute $\mathcal{R}(b)$ from $\mathcal{A}(O_{a^r})$ in linear-time.

Lemma 37. *Let a be an arc of T_b connected to a leaf and let c be the corresponding child of b in $\mathcal{B}$. Then, $\mathcal{A}(O_a)$ can be computed in linear-time from $\mathcal{R}(c)$.*

Proof. First note that $mid(a) = \{s_c, t_c\}$, where (s_c, t_c) is the reference edge of c in $\mathcal{B}$. Moreover, because of Lemma 10, we can assume that $O_a = N_c$ since their subcurves connect the same two vertices in the same face. Therefore, there is a one-to-one correspondence between the types in $\mathcal{R}(c)$ and the types in $\mathcal{A}(O_a)$. Moreover, given a type $X = (\psi, M, S) \in \mathcal{R}(c)$, then the corresponding type $X' = (\psi', M', S')$ in $\mathcal{A}(O_a)$ can be obtained as follows. Let $\alpha : \{L, R\} \to O_a\}$ the bijection such that $\alpha(L) = c$ if c is equal to L_c and $\alpha(c) = R$ otherwise.

- We define ψ' by setting $\psi'(\alpha(A)) = \emptyset$ if $\psi(A) = \emptyset$, $\psi'(\alpha(A)) = [x]$ if $|\psi(A)| = 1$, and $\psi'(\alpha(A)) = [x, x']$ if $|\psi(A)| = 2$ for every $A \in \{L, R\}$,
- M' is obtained from M after replacing the vertices in $\psi(A)$ with their counterparts in $\psi'(\alpha(A))$ for every $A \in \{L, R\}$,
- $S' = S$.

Therefore, we obtain $\mathcal{A}(O_a)$ as the set $\{\, X' \mid X \in \mathcal{R}(c) \,\}$, which also shows that it can be computed in linear-time from $\mathcal{R}(c)$. ◻

Lemma 38. *$\mathcal{R}(b)$ can be computed in linear-time from $\mathcal{A}(O_{a^r})$.*

Proof. First note that $mid(a^r) = \{s_b, t_b\}$ and therefore that O_{a^r} consists of two subcurves $c = (\{s, t\}, f)$ and $c' = (\{s, t\}, f')$, where both f and f' have the reference edge (s_b, t_b) on their border. Therefore, every type $X = (\psi, M, S) \in \mathcal{A}(O_{a^r})$ can be easily translated into two types of b after specifying a bijection α between $\{L, R\}$ and $\{c, c'\}$. That is given such a bijection $\alpha : \{L, R\} \to \{c, c'\}$, we obtain the type $X_\alpha = (\psi_\alpha, M_\alpha, S_\alpha)$ of b corresponding to X by setting:

- $\psi_\alpha(L) = \emptyset$ if $\psi(\alpha(L)) = \emptyset$, $\psi_\alpha(L) = \{l\}$ if $|\psi(\alpha(L))| = 1$, and $\psi_\alpha(L) = \{l, l'\}$ if $|\psi(\alpha(c))| = 2$,
- M_α is obtained from M by replacing the first vertex in $\psi(\alpha(L))$ ($\psi(\alpha(R))$) with l (r) and the second vertex in $\psi(\alpha(L))$ ($\psi(\alpha(R))$) with l' (r'),
- $S_\alpha = S_\alpha$.

It is now straightforward to verify that O_{a^r} has type X if and only if b has type X_{α_1} and X_{α_2} for the two possible bijections α_1 and α_2 between $\{L, R\}$ and $\{c, c'\}$. Therefore, it holds that $\mathcal{R}(b) = \{ X_{\alpha_1}, X_{\alpha_2} \mid X \in \mathcal{A}(O_{a^r}) \}$, which shows that $\mathcal{R}(b)$ can be computed in linear-time from $\mathcal{A}(O_{a^r})$. ◻

Given the above Lemmas, it now merely remains to show how to compute $\mathcal{A}(O_{a_P})$ from $\mathcal{A}(O_{a_L})$ and $\mathcal{A}(O_{a_R})$ for any inner node of T_b with parent arc a_P and child arcs a_L and a_R. Employing our framework introduced in Subsection 4.3 allows us to solve a simpler problem instead, i.e., we only have to show how to compute $\mathcal{A}(O_1 \oplus O_2)$ from $\mathcal{A}(O_1)$ and $\mathcal{A}(O_2)$ for any weak nooses O_1 and O_2.

Let O_1 and O_2 be two weak nooses having type $X_1 = (\psi_1, M_1, S_1)$ and type $X_2 = (\psi_2, M_2, S_2)$, respectively. We say that X_1 and X_2 are *compatible* if

(1) $O = O_1 \oplus O_2$ is a weak noose,
(2) the inside region of the noose O contains all subcurves in $(O_1 \cap O_2)$,
(3) $\forall c \in O_1 \cap O_2$, it holds $\psi_1(c) = \psi_2(c)$,
(4) for every $u \in V(O_1 \cap O_2) \setminus V(O_1 \oplus O_2)$, it holds that u is only in one of following sets: S_1, S_2 or $V(M_1) \cap V(M_2)$, and
(5) the multi-graph obtained from the union of M_1 and M_2 is acyclic, or is one cycle and $V(O) \subseteq S_1 \cup S_2 \cup (V(M_1) \cap V(M_2))$,
(6) if X_1 is the full type, then X_2 is the empty type and $V(O_2) \subseteq V(O_1)$, and vice versa.

Please also refer to Figure 7 for an illustration of two compatible types. Let $X_1 = (\psi_1, M_1, S_1)$ and $X_2 = (\psi_2, M_2, S_2)$ be two compatible types defined on weak nooses O_1 and O_2, respectively.

We denote by $X_1 \circ X_2$ the *combined type* $X = (\psi, M, S)$ of $X_1 = (\psi_1, M_1, S_1)$ and $X_2 = (\psi_2, M_2, S_2)$ for the weak noose $O = O_1 \oplus O_2$ that is defined as follows. For each $c \in O$, if $c \in O_1$ then $\psi(c)$ is equal to $\psi_1(c)$, otherwise $\psi(c)$ is equal to $\psi_2(c)$ and the set S is equal to $(S_1 \cup S_2 \cup (V(M_1) \cap V(M_2))) \cap V(O)$, i.e., any vertex with degree two w.r.t. X must be in $V(O)$ and have degree two already w.r.t. X_1 or X_2, or it must be in both matchings M_1 and M_2. If either X_1 or X_2 is a full type, then by *(6)* we get that $M_1 = M_2 = M = \emptyset$ and $X_1 \circ X_2$ is the full type. If the multi-graph $M_1 \cup M_2$ is one cycle, then by *(5)* we get that $M = \emptyset$ and $X_1 \circ X_2$ is the full type. Otherwise, due to *(5)*, the multi-graph $M_1 \cup M_2$ is acyclic and corresponds to a set of paths. Therefore, the matching M is the set containing the two endpoints for every path in $M_1 \cup M_2$.

Observation 39. *Let X_1 and X_2 be two types defined on the weak nooses O_1 and O_2, respectively. Then, we can check whether X_1 and X_2 are compatible and if so compute the type $X_1 \circ X_2$ in time $O(|O_1| + |O_2|)$.*

The following two lemmas are crucial for showing the correctness of our approach. The former shows that if there is a witness W for G that respects $\mathcal{B}$, then for every two weak nooses O_1 and O_2 it holds that $\Gamma_W(b, O_1)$ and $\Gamma_W(b, O_2)$ are compatible types and $\Gamma_W(b, O) = \Gamma_W(b, O_1) \circ \Gamma_W(b, O_2)$. The latter shows in some sense the reverse direction, i.e., if O_1 and O_2 have compatible types X_1 and X_2, then $O = O_1 \oplus O_1$ has type $X_1 \circ X_2$.

Lemma 40. *Let $W = (D, D_H, G_H, H)$ be a witness for G that* respects *$\mathcal{B}$. Let b be an R-node or an S-node with sphere-cut decomposition $\langle T_b, \lambda_b, \Pi_b \rangle \in \mathcal{T}$. Let O_1 and O_2 be two weak nooses that are subsets of $C(T_b)$ and satisfy properties* (1) *and* (2)*, i.e., $O = O_1 \oplus O_2$ is also a weak noose and the inside region of O contains all subcurves in $(O_1 \cap O_2)$. Then, $X_1 = \Gamma_W(b, O_1)$ and $X_2 = \Gamma_W(b, O_2)$ are compatible types and $\Gamma_W(b, O) = X_1 \circ X_2$.*

Proof. Let $i \in \{1, 2\}$ and let $X_i = (\psi_i, M_i, S_i)$ be the type $\Gamma_W(b, O_i)$. The properties *(1)* and *(2)* given in the description of compatible types are a direct consequence of the assumptions of this lemma. Recall that H^{O_i} is defined in Subsection 4.3 and essentially corresponds to the subgraph of H including crossings at the subcurves of O_i inside O_i. If H^{O_i} contains a cycle then X_i and X are the full types. Also, for $j \in \{1, 2\} \setminus \{i\}$, X_j must be the empty type and $V(O_j) \subseteq V(H^{O_i})$, which satisfy property *(6)*. In this case the properties *(3)*, *(4)* and *(5)* are satisfied, because $V(\psi_i) = V(\psi_j) = M_1 = M_2 = \emptyset$ and $S_i = V(O_i)$.

Otherwise, let $\mathcal{P}_i$ be a set of all maximal paths in H^{O_i} each of size at least 2. Then, $(\mathcal{P}_i, D_H^{O_i})$ witnesses that O_i has type X_i. Note that $\bigcup(\mathcal{P}_1 \cup \mathcal{P}_2)$ is almost equal to H^O. In fact, H^O only misses the vertices in $V(\psi_1) \cap V(\psi_2)$. We therefore define H_*^O as the graph obtained from H^O after subdividing the edges that cross the subcurves in $O_1 \cap O_2$. Then, we can assume that $\bigcup(\mathcal{P}_1 \cup \mathcal{P}_2)$ is equal to H_*^O.

The property *(3)* is simply obtained from the fact that X_1 and X_2 are obtained from the same Hamiltonian cycle and therefore agree on all subcurves shared between O_1 and O_2.

For each v in $V(O_1 \cap O_2) \setminus V(O)$, v has degree 2 in H^O, because v is not in $V(O)$. Since $\bigcup(\mathcal{P}_1 \cup \mathcal{P}_2) = H_*^O$, it follows that v is in one of the following sets: $\mathrm{IN}(\mathcal{P}_1) \cap V(O_1)$, $\mathrm{IN}(\mathcal{P}_2) \cap V(O_2)$ and $\big(V(\mathcal{P}_1) \setminus \mathrm{IN}(\mathcal{P}_1)\big) \cap \big(V(\mathcal{P}_2) \setminus \mathrm{IN}(\mathcal{P}_2)\big)$, which correspond to sets S_1, S_2 and $V(M_1) \cap V(M_2)$, respectively. This demonstrates property *(4)* .

Property *(5)* now follows because the matchings M_1 and M_2 have an edge between the endpoints of every path in $\mathcal{P}_1$ and $\mathcal{P}_2$, so if H_*^O is acyclic then the multi-graph $M_1 \cup M_2$ is also acyclic, otherwise H_*^O is a cycle and the multi-graph $M_1 \cup M_2$ is a cycle, and $V(O) \subseteq V(H_*^O)$. □

Lemma 41. *If O_1 and O_2 have compatible types X_1 and X_2, respectively, then $O = O_1 \oplus O_2$ has type $X = X_1 \circ X_2$.*

Proof. Let $X_1 = (\psi_1, M_1, S_1)$, $X_2 = (\psi_2, M_2, S_2)$, and $X = (\psi, M, S)$. For each $i \in \{1, 2\}$, let $(\mathcal{P}_i, D_i)$ be the witness that O_i has type X_i. Since X_1 and X_2 are compatible and in particular because of properties *(2)* and *(3)* from the definition of compatible types, the drawing $D = D_1 \cup D_2$ is planar. Note that O is a weak noose because of property *(1)*. Consider the graph $H = \bigcup(\mathcal{P}_1 \cup \mathcal{P}_2)$. Because of property *(4)* all endpoints of the paths in H are in $V(O) \cup V(\psi)$.

If H is a single cycle then from the property *(5)* and *(6)*, we get that $X_1 \circ X_2$ is a full type and the witness is (H, D).

Otherwise, the multi-graph $M_1 \cup M_2$ is a disjoint union of paths, due to the property *(5)*. Note that each path in $\mathcal{P}_i$ corresponds to an edge in M_i, for $i \in \{1, 2\}$. So H is also a disjoint union of paths and let $\mathcal{P}$ be the set of paths of H. Then $(\mathcal{P}, D)$ is the witness of that O has the type X. □

The following lemma is required to compute the types for a weak noose $O = O_1 \oplus O_2$ and provides a detailed analysis of the run-time required.

Lemma 42. *Let O, O_1 and O_2 be weak nooses such that $O = O_1 \oplus O_2$. There are at most $6(84\sqrt{14})^k$ triples (X, X_1, X_2) such that X, X_1 and X_2 are types defined on O, O_1 and O_2, respectively, X_1 is compatible with X_2, and $X = X_1 \circ X_2$. Moreover, all such triples can be enumerated in $O((84\sqrt{14})^k k)$, where $k = max\{|O|, |O_1|, |O_2|\}$.*

Proof. We define the role of a vertex or subcurve in a type as the information stored in the type about that vertex or subcurve. First, we will show that for fixed X the role of each vertex $V(O) \setminus V(O_1 \cap O_2)$ and each subcurve from O in types X_1 and X_2 remains the same. Let $X = (\psi, M, S)$ be a type that can be defined on a weak noose O and u be an arbitrary vertex from $V(O) \cap V(O_1) \cap V(O_2)$. For each $i \in \{1, 2\}$, let v_i be first vertex after u in clockwise orientation such that $v_i \in V(M) \setminus V(O_1 \cap O_2)$ and v_i is a vertex from the subcurves from the segment $O \cap O_i$. Let DW^i be the Dyck word corresponding to the matching M from Observation 11 with starting vertex v_i and clockwise orientation and let $DW^i_\star$ be a prefix of DW^i corresponding to vertices on the subcurves from the segment $O \cap O_i$. Let $X_1 = (\psi_1, M_1, S_1)$ and $X_2 = (\psi_2, M_2, S_2)$ be the types that can be defined on O_1 and O_2 respectively, such that $X = X_1 \circ X_2$. Let DW_i be the Dyck word corresponding to the matching M_i from Observation 11 with starting vertex v_i and clockwise orientation. Note that for different pairs X_1 and X_2, the type X remains the same and therefore also $DW^1_\star$, $DW^2_\star$ and S. Moreover, $DW^i_\star$ is a prefix of DW_i and $S_i \cap (V(O \cap O_i) \setminus V(O_1 \cap O_2)) = S \cap (V(O \cap O_i) \setminus V(O_1 \cap O_2))$. This means that, the role of each vertex $V(O \cap O_i) \setminus V(O_1 \cap O_2)$ and each subcurve $O \cap O_i$ in type X_i remains the same, so the only places where the different pairs of compatible types may differ is in the segment $O_1 \cap O_2$.

Secondly, we will bound number of different pairs X_1 and X_2 for fixed X. By condition *(1)* from the definition of compatible types, for each $v \in V(O_1 \cap O_2) \setminus V(O)$, there are $2 + 2 \cdot 2 = 6$ different combinations of roles of v in types X_1, X_2, i.e., $v \in S_1 \land v \notin S_2 \cup V(M_2)$, $v \in S_2 \land v \notin S_1 \cup V(M_1)$ or $v \in V(M_1) \cap V(M_2)$ and v corresponds to either "[" or "]" in DW_1 and either "[" or "]" in DW_2. Moreover, for each $v \in V(O_1 \cap O_2) \cap V(O)$ there are at most 6 different combination of roles of v in types X_1 and X_2, because the role of v in type X is known and this can only decrease number of combinations. Due to the type definition and condition *(3)* from the definition of compatible types, we obtain that $\bigcup_{c \in O_1 \cap O_2} \psi_1(c) = \bigcup_{c \in O_1 \cap O_2} \psi_2(c) = V(\psi_1) \cap V(\psi_2) \subseteq V(M_1) \cap V(M_2)$. Therefore, for each $v \in V(\psi_1) \cap V(\psi_2)$, there are $2 \cdot 2 = 4$ different combinations of roles of v in types X_1 and X_2, i.e., v corresponds to either "[" or "]" in DW_1 and either "[" or "]" in DW_2. For each subcurve $c \in O_1 \cap O_2$, there are 3 possible values $\{\emptyset, [x], [x, x']\}$ for $\psi_1(c)$, and therefore there are $1 + 4 + 16 = 21$ possibilities, i.e., 1, 4, and 16 possibilities in case that $\psi_1(c) = \emptyset$, $\psi_1(c) = [x]$, and $\psi_1(c) = [x, x']$, respectively, of the role of c in types X_1 and X_2. Furthermore, since $|V(O_1 \cap O_2)| = |O_1 \cap O_2| + 1$, there are at most $6^{|O_1 \cap O_2|+1} 21^{|O_1 \cap O_2|} = 6 \cdot 126^{|O_1 \cap O_2|}$ different pairs of types X_1 and X_2 for fixed X.

There are at most $28^{|O|}$ different types X that can be defined on O, due to the Lemma 19, and there are $6 \cdot 126^{|O_1 \cap O_2|}$ different pairs of types X_1 and X_2 that can be defined on O_1 and O_2 respectively, such that $X = X_1 \circ X_2$, so there are at most $28^{|O|} \cdot 6 \cdot 126^{|O_1 \cap O_2|}$ different triples (X, X_1, X_2). Note that $|O_1| + |O_2| - 2|O_1 \cap O_2| = |O|$ therefore there are $\mathcal{O}(28^k 126^{\frac{k}{2}}) = \mathcal{O}(84\sqrt{14}^k)$ different triples (X, X_1, X_2).

In order to generate all valid triples, first we enumerate all possible types X, of which there are $\mathcal{O}(28^{|O|})$, in time $\mathcal{O}(28^{|O|}|O|)$ using Lemma 19. Then based on type X we fix the role of all vertices in $V(O) \setminus V(O_1 \cap O_2)$ and all subcurves in O in types X_1 and X_2. We can then assign a role to each vertex in $V(O_1 \cap O_2)$ and every subcurve of $O_1 \cap O_2$ for types X_1 and X_2 and verify that the corresponding words are Dyck words in time $\mathcal{O}(|O_1| + |O_2|)$ and if so translate it into a type description using Observation 11. Lastly we check if X_1 and X_2 are compatible and if so check if $X = X_1 \circ X_2$, in $\mathcal{O}(|O_1| + |O_2|)$ time using Observation 39. Therefore the time complexity of this operation is $\mathcal{O}(28^{|O|}(|O| + 126^{|O_1 \cap O_2|} \cdot (|O_1| + |O_2|))) = \mathcal{O}(84\sqrt{14}^k k)$, due to equation $|O_1| + |O_2| - 2|O_1 \cap O_2| = |O|$. □

Lemma 43. *Let b be an R-node or S-node and let a_P be a parent arc with two child arcs a_L and a_R in the sphere-cut decomposition $\langle T_b, \lambda_b, \Pi_b \rangle$ of $\textsc{Sk}(b)$. We can compute $\mathcal{A}(O_{a_P})$ from $\mathcal{A}(O_{a_L})$ and $\mathcal{A}(O_{a_R})$ in $\mathcal{O}(315^k)$ time, where $k = max(|mid(a_P)|, |mid(a_L)|, |mid(a_R)|)$.*

Proof. Note first that $\textsc{Sk}(b)$ is biconnected because b is either an R-node or an S-node. Therefore, we can apply Lemma 18, to obtain a sequence Q of at most 3 $\oplus$-operations such that:

- Q contains only the weak nooses O_{a_L}, O_{a_R} and at most two weak nooses O^1 and O^2 each bounding an edge-less graph with three vertices.

- Every step of Q produces a weak noose O such that $|O| \leq 1 + k$ and O_{a_P} is the weak noose produced by Q after the final step.

Before we can employ Q to compute $\mathcal{A}(O_{a_P})$, we first need to compute $\mathcal{A}(O^i)$ for the at most two weak nooses O^1 and O^2. To do so we employ Lemma 19 to enumerate all possible types X of O^i, which because $|V(O^i)| \leq 3$ can be achieved in constant time. We then add each of those types to $\mathcal{A}(O^i)$; this is correct because the noose does not contain any edges and therefore allows for every possible type. We then compute $\mathcal{A}(O_{a_P})$ using Q as follows. For every step of Q, which given two weak nooses O_1 and O_2 for which the set of types $\mathcal{A}(O_1)$ and $\mathcal{A}(O_2)$ have already been computed, computes the weak noose $O = O_1 \oplus O_2$, we do the following to compute $\mathcal{A}(O)$. Let $k' = \max\{|O|, |O_1|, |O_2|\}$. Using Lemma 42 we enumerate all of the at most $6(84\sqrt{14})^{k'}$ triples (X, X_1, X_2) of types defined on O, O_1, O_2, respectively, in time $O((84\sqrt{14})^{k'} k')$. Then, for each such triple (X, X_1, X_2), we check (in constant time) whether $X_1 \in \mathcal{A}(O_1)$ and $X_2 \in \mathcal{A}(O_2)$ and if so we add X to $\mathcal{A}(O)$. Because $k' \leq k + 1$ and since Q consists of at most 3 steps, we obtain that computing all steps of Q and therefore computing the set $\mathcal{A}(O_{a_P})$ takes time at most $O((84\sqrt{14})^{k+1}(k+1)) = O(315^k)$. Finally, the correctness of the procedure follows immediately from Lemmas 40 and 41. □

Proof of Lemma 36. We first use Lemma 8 to compute a sphere-cut decomposition $\langle T_b, \lambda_b, \Pi_b \rangle$ of $\textsc{Sk}(b)$, whose width ω is equal to the branchwidth of G, having at most $O(|V(\textsc{Sk}(b)|) = O(\ell)$ nodes in time $O(\ell^3)$. Note that to compute $\langle T_b, \lambda_b, \Pi_b \rangle$ we can use any of the (at most) two planar drawings of $\textsc{Sk}(b)$ that contain the reference edge (s_b, t_b) in the outer-face, since we will take the resulting symmetries into account when we compute the set of types; more specifically in Lemma 38 and Lemma 37.

We then compute $\mathcal{A}(O_{a^r})$ using a bottom-up dynamic programming algorithm on T_b. In particular, we use Lemma 37 to compute $\mathcal{A}(O_a)$ for all arcs in T_b incident to a leaf node of T_b and then we use Lemma 43 to compute $\mathcal{A}(O_a)$ for any other arc a of T_b in a bottom-up manner. Having computed $\mathcal{A}(O_{a^r})$, we then use Lemma 38 to obtain $\mathcal{R}(b)$ from $\mathcal{A}(O_{a^r})$. The correctness of the algorithm follows from the employed lemmas. To analyze the run-time of the algorithm, we first note that we require time at most $O(\ell^3)$ to compute the sphere-cut decomposition $\langle T_b, \lambda_b, \Pi_b \rangle$. Moreover, the run-time of the dynamic programming algorithm on $\langle T_b, \lambda_b, \Pi_b \rangle$ is at most equal to the number of inner nodes of T_b, i.e., at most $|E(\textsc{Sk}(b))| = \ell + 1$, times the time required for one application of Lemma 43, i.e., at most $O(315^\omega)$, where ω is the width of T_b; note that here we use that $k = max\{|mid(a_P)|, |mid(a_L)|, |mid(a_R)|\} \leq \omega$. Therefore, we obtain $O(315^\omega \ell + \ell^3)$ as the total run-time required to compute $\mathcal{R}(b)$. □

5.3 Putting Everything Together

Here, we put everything together and prove Lemma 22. Before doing so, we first need the following simple lemma that allows us to compute the set of types for every leaf node of $\mathcal{B}$ in constant time.

Lemma 44. *Let l be a leaf-node (and Q-node) of $\mathcal{B}$. We can compute $\mathcal{R}(l)$ in time $O(1)$.*

Proof. Let l be a leaf-node with reference edge (s, t) of $\mathcal{B}$. Then, l is also a Q-node with edge $\{s, t\}$ due to the properties of SPQR-trees. Let $\psi_{x,y}$ for $x, y \in [0, 2]$ be defined by setting $\psi_{0,y}(L) = \emptyset$, $\psi_{1,y}(L) = \{l\}$, $\psi_{2,y}(L) = \{l, l'\}$, $\psi_{x,0}(R) = \emptyset$, $\psi_{x,1}(R) = \{r\}$, and $\psi_{x,2}(R) = \{r, r'\}$.

$\mathcal{R}(l)$ contains the following types:

- Types for $\psi_{0,0}$:
  - the type $(\psi_{0,0}, \emptyset, \{s, t\})$ indicating a Hamiltonian cycle on $\textsc{Pe}(l)$,
  - the type $(\psi_{0,0}, \emptyset, \emptyset)$,

  - the type $(\psi_{0,0}, \{\{s,t\}\}, \emptyset)$;
- Types for $\psi_{1,0}$ (symmetrically for $\psi_{0,1}$):
  - the types $(\psi_{1,0}, \{\{l,s\}\}, \emptyset)$ and $(\psi_{1,0}, \{\{l,s\}\}, \{t\})$,
  - the types $(\psi_{1,0}, \{\{l,t\}\}, \emptyset)$ and $(\psi_{1,0}, \{\{l,t\}\}, \{s\})$;
- Types for $\psi_{1,1}$:
  - for every $S \subseteq \{s,t\}$ the type $(\psi_{1,1}, \{\{l,r\}\}, S)$,
  - the types $(\varphi_{1,1}, \{\{l,s\},\{t,r\}\}, \emptyset)$ and $(\varphi_{1,1}, \{\{l,t\},\{s,r\}\}, \emptyset)$;
- Types for $\psi_{2,0}$ (symmetrically for $\psi_{0,2}$):
  - for every $S \subseteq \{s,t\}$, the type $(\psi_{2,0}, \{\{l,l'\}\}, S)$,
  - the type $(\psi_{2,0}, \{\{s,t\},\{l,l'\}\}, \emptyset)$,
  - the type $(\psi_{2,0}, \{\{l,s\},\{l',t\}\}, \emptyset)$,
- Types for $\psi_{2,1}$ (symmetrically for $\psi_{1,2}$):
  - the type $(\psi_{2,1}, \{\{l,r\},\{l',t\}\}, \{s\})$,
  - for every $S \in \{\emptyset, \{t\}\}$, the type $(\psi_{2,1}, \{\{l,l'\},\{s,r\}\}, S)$,
  - for every $S \in \{\emptyset, \{s\}\}$, the type $(\psi_{2,1}, \{\{l,l'\},\{t,r\}\}, S)$,
- Types for $\psi_{2,2}$:
  - the type $(\psi_{2,2}, \{\{l,l'\},\{s,r\},\{t,r'\}\}, \emptyset)$,
  - the type $(\psi_{2,2}, \{\{r,r'\},\{l,s\},\{l',t\}\}, \emptyset)$,
  - the type $(\psi_{2,2}, \{\{l,l'\},\{r,r'\},\{s,t\}\}, \emptyset)$,
  - for every $S \subseteq \{s,t\}$, the type $(\psi_{2,2}, \{\{l,l'\},\{r,r'\}\}, S)$,
  - the type $(\psi_{2,2}, \{\{l,r\},\{l',r'\}\}, \{s,t\})$,

Note that $\mathcal{R}(l)$ can be computed in constant time and actually contains all types of $\text{Pe}(l)$ and therefore also satisfies *(R1)* and *(R2)*. □

We are now ready to show Lemma 22.

Lemma 22. *Let G be a biconnected multi-graph with n vertices and m edges and SPQR-tree $\mathcal{B}$. Then, we can decide in time $O(315^\omega n + n^3)$ whether G is subhamiltonian, where ω is the maximum branchwidth of $\text{Sk}(b)$ over all R-nodes and S-nodes b of $\mathcal{B}$.*

Proof. We start by showing how to compute the set of types $\mathcal{R}(b)$ for every node b of the SPQR-tree $\mathcal{B}$, which we will achieve using a bottom-up dynamic programming algorithm along $\mathcal{B}$. As stated in Section 2, we assume that $\mathcal{B}$ is rooted at some Q-node with edge e, whose child b_r has e as its reference edge. Starting at the leaves of $\mathcal{B}$, we use Lemma 44 to compute $\mathcal{R}(l)$ for every leaf node l of $\mathcal{B}$ in constant time. We then iteratively consider the inner nodes b for which $\mathcal{R}(c)$ for all children c of b in $\mathcal{B}$ have already been computed. Let b be a node of $\mathcal{B}$ with ℓ children. If b is an R-node or an S-node, we use Lemma 36 to compute $\mathcal{R}(b)$ in time $O(315^\omega \ell + \ell^3)$, where ω is the branchwidth of $\text{Sk}(b)$. Otherwise b is a P-node and we use Lemma 23 to compute $\mathcal{R}(b)$ in time $O(\ell))$. By applying the above procedure exhaustively, we obtain the set $\mathcal{R}(b)$ of types for all nodes apart from the root node r of $\mathcal{B}$; this is because r is a Q-node which is not a leaf of $\mathcal{B}$. Let b_r be the unique child of r in $\mathcal{B}$ and let $e = (s,t)$ be the reference edge of b_r (which is also the reference edge of r, because r is a Q-node). Since b_r is not the root of $\mathcal{B}$, we have computed the set $\mathcal{R}(b_r)$ of types for b_r. We now claim that G is subhamiltonian if and only if $(\psi_\emptyset, \emptyset, \{s,t\}) \in \mathcal{R}(b_r)$, where $\psi_\emptyset(L) = \psi_\emptyset(R) = \emptyset$. Towards showing the forward direction of the claim suppose that G is subhamiltonian. It then follows from Lemma 16 that G

has a witness (D, D_H, G_H, H) that respects $\mathcal{B}$. Consequently, we obtain from *(R2)* that $\Gamma_W(b_r) \in \mathcal{R}(b_r)$. Therefore, if $\Gamma_W(b_r) = (\psi_\emptyset, \emptyset, \{s,t\})$, then we are done. Otherwise, consider first the case that H contains the edge $\{s,t\}$ of G. In this case we can replace the edge in H by adding a new edge between s and t, which we can draw arbitrary close to the original edge in G between s and t. Therefore, we can assume that H does not contain the edge of G between s and t. But then, we can obtain a new witness $W' = (D, D'_H, G_H, H)$ that respects $\mathcal{B}$ such that $\Gamma_{W'}(b_r) = (\psi_\emptyset, \emptyset, \{s,t\})$ by changing the drawing D_H of H into the new drawing D'_H such that H touches the noose N_{b_r} only at s and t; this can be achieved by replacing every subcurve in D_H of H outside of N_{b_r} with a curve inside N_{b_r} drawn arbitrarily close to N_{b_r}. Towards showing the reverse direction, suppose that $(\psi, \emptyset, \{s,t\}) \in \mathcal{R}(b_r)$. By the definition of a type, it follows that there is a Hamiltonian cycle H for G that can be drawn together with $G \setminus \{e\}$ entirely within the noose N_b. But then, H can also be drawn together with G and therefore shows that G is subhamiltonian, as required.

The run-time of the algorithm is at most the number of nodes of $\mathcal{B}$, which because of Lemma 5 is at most $O(|E(G)|) = O(|V(G)|)$ (because G is planar), times the maximum time required by the dynamic programming procedure at every node of $\mathcal{B}$. Since the latter is dominated by the time required for R-nodes, i.e., $O(315^\omega + \ell^3)$ due to Lemma 36, where ω is the branchwidth of $\textsc{Sk}(b)$ and ℓ is the number of children of the node in $\mathcal{B}$, we obtain $O((315^\omega |V(G)| + |V(G)|^3))$ as the total run-time of the algorithm. ◻

6 An Algorithm Using the Feedback Edge Number

In this section, we establish the following theorem:

Theorem 45. Book Thickness *is fixed-parameter tractable when parameterized by the feedback edge number of the input graph.*

To obtain the result, we distinguish whether the bound on the number of pages is 2, or more. We begin with the latter case.

6.1 The Case with More than Two Pages

The remainder of this section is devoted to a proof of Theorem 46 (stated below), which is based on providing an (exponentially sized) kernel for the problem. We begin by introducing a few section-specific definitions.

Theorem 46. *When restricted to inputs (G, k) such that $k \geq 3$,* Book Thickness *is fixed-parameter tractable parameterized by the feedback edge number.*

Notation and Definitions. Let G be an n-vertex graph and $L = (\prec, \sigma)$ be a k-page embedding of G. For the purposes of this section, it will be useful to think of the linear order $\prec$ on $V(G) = \{v_1, \dots, v_n\}$ as a set of n points on the real line such that $v_1 < v_2 < \dots < v_n$. With this interpretation in mind, we can define a *region* as the open interval between two consecutive elements of $V(G)$.

Let $U \subseteq V(G)$ be a subset of vertices of G. We say that a path P of G is *maximal proper for* U if P (1) is not only a single edge, (2) every internal vertex v of P satisfies $\deg_G(v) = 2$ and $v \notin U$, and (3) P is maximal (with respect to containment) among all paths satisfying properties (1) and (2).

Let $L = (\prec, \sigma)$ be a book embedding, we say that edge $e' = u'v'$ is *nested* under the edge $e = uv$ if $\sigma(e) = \sigma(e')$ and $(u', v') \subset (u, v)$, that is either $u \prec u' \prec v' \prec v$ or $u = u' \prec v' \prec v$ or $u \prec u' \prec v' = v$. Moreover, we say that an edge *touches* a region if one of its endpoints belongs to the interior of that region, and that a path *touches* a region t times if t

of its edges touches that region. We use $\sigma(v) = \{\sigma(e) \mid e \in E(G) \wedge v \in e\}$ to denote the set of all the page numbers of the edges incident to v.

We say that a graph G and a set of paths $\mathcal{P}$ are *near-disjoint* if the vertices that are in common among G and the paths of $\mathcal{P}$ are exactly their endpoints. Let G be a graph and $\mathcal{P}$ be a set of paths such that G and $\mathcal{P}$ are near-disjoint: we say that a set of paths $\mathcal{P}'$ *serves as* $\mathcal{P}$ if G and $\mathcal{P}'$ are near-disjoint, $|\mathcal{P}'| = |\mathcal{P}|$, and there is a path in $\mathcal{P}$ with u and v as endpoints if and only if there is a path in $\mathcal{P}'$ with u and v as endpoints. If a graph G and a set of paths $\mathcal{P}$ are near-disjoint, we denote the graph obtained by inserting P into G as $G \vee \mathcal{P}$.

The Branching Step. In the first part of the algorithm, we apply brute-force branching and a simple preprocessing rule. We begin by exhaustively removing all pendant vertices in the graph.

Lemma 47. *Let G be a graph and v be a vertex of degree at most one in G. Then (G, k) and $(G - v, k)$ are equivalent instances of* Book Thickness.

Proof. We show that (G, k) is a yes-instance of Book Thickness if and only if $(G - v, k)$ is a yes-instance of the same problem. The forward direction follows directly from the fact YES instances are preserved when considering subgraphs: if H is a subgraph of G, a k-page book embedding of H can be obtained from a k-page book embedding of G by removing $G - H$.

Consider now a k-page book embedding L for $G - v$. If v is an isolated vertex of G, a k-page book embedding for G can be obtained from L by inserting v in any region of L. Suppose v is a leaf of G and let u_v be the unique neighbor of v in G. A k-page book embedding L' for G can be obtained from L by inserting v in any of the two regions of L adjacent to u_v and by setting $\sigma(vu_v) = 1$. Note that L' is indeed a k-page book embedding because the edge vu_v neither nests nor intersects any other edge (on any page). □

Let G be a graph and let $G_{\geq 2}$ be the graph obtained from G by removing, exhaustively, every vertex of degree at most one. Note that every vertex of $G_{\geq 2}$ has at least two neighbors.

Corollary 48 (of Lemma 47)**.** *Then instances (G, k) and $(G_{\geq 2}, k)$ are equivalent instances of* Book Thickness.

Let F be a minimum feedback edge set of $G_{\geq 2}$; note that $|F| = \text{fen}(G) = \text{fen}$, since none of the vertices or edges that are present in G but not in $G_{\geq 2}$, that is no element of $(V(G) \setminus V(G_{\geq 2})) \cup (E(G) \setminus E(G_{\geq 2}))$, is part of a cycle of G. We define G_F to be the tree $G_{\geq 2} - F$. We denote with T_1 and $T_{\geq 3}$ the set of leaves and of vertices of degree at least 3 of G_F, respectively. The elements of $T_{\geq 3}$ are also usually called *branching vertices*.

Let V_F be the set of all the vertices in G_F that are adjacent to an edge of F in $G_{\geq 2}$, and note that $T_1 \subseteq V_F$. Let $B_F = T_{\geq 3} \cup V_F$ and let $\mathcal{P}_F$ be the set of all maximal proper paths of G_F for B_F.

Lemma 49. **(a)** $|T_{\geq 3}| \leq |T_1| \leq |V_F| \leq 2\text{fen}$ *and* **(b)** $|\mathcal{P}_F| \leq |B_F| \leq 4\text{fen}$.

Proof. Let us first prove that $|T_{\geq 3}| \leq |T_1|$. It is immediate to note that in any tree the number of branching vertices is at most the number of leaves. Recall that $T_1 \subseteq V_F$. Finally recall is V_F is defined as the set of all the vertices in $G_{\geq 2}$ that are adjacent to an edge of F and so the number of vertices in V_F is at most twice the size of F. This completes the proof of **(a)**.

By recalling that $B_F = T_{\geq 3} \cup V_F$, we obtain that $|B_F| \leq 4\text{fen}$. Now let us consider the auxiliary graph G'_F that is obtained from G_F by contracting to one edge every path of $\mathcal{P}_F$.

Note that G'_F is a tree and there are two 1-to-1 correspondences: one between the vertices of G'_F and B_F, and one between the edges of G'_F and $\mathcal{P}_F$. It follows that $|\mathcal{P}_F| = |E(G'_F)| \leq |V(G'_F)| - 1 \leq |B_F|$. This proves **(b)**. □

We remark that, in view of Fact 1, all of these objects are efficiently computable.

Observation 50. *It is possible to compute $G_{\geq 2}$, a minimum feedback edges set F of $G_{\geq 2}$, G_F, T_1, $T_{\geq 3}$, V_F, B_F and $\mathcal{P}_F$ in polynomial time.*

Thanks to Observation 50, we are able to compute B_F, $\mathcal{P}$ and, by Lemma 49, all these sets have bounded size.

At this point, the only issue to obtain a kernel of the desired size is that $\mathcal{P}_F$ might contain paths of unbounded length. Long Path Insertion. As our first step towards dealing with the long paths that remain in our instance, we show that yes-instances are preserved if we extend a path that is near-disjoint with the rest of the graph.

Lemma 51. *Let G be a graph and P be a path of length at least 2 such that G and $\{P\}$ are near-disjoint. If $G \vee P$ admits a k-page book embedding, then $G \vee P'$ admits a k-page book embedding, where P' is obtained from P by subdividing an arbitrary edge of P once.*

Proof. Consider a k-page book embedding L for $G \vee P$ and let $e = uv$ of be an edge of P. Now we want to show that we can always subdivide e once. First, suppose there is no edge that is nested under e. In this case, we can delete e, add a vertex w such that $u \prec w \prec v$, and insert the edges uw and wv.

Since P is of length at least two, one of the two endpoints of e is an internal vertex of P, say it is u (the case for v can be proven symmetrically). Consider any edge $e' = u'v'$ that is nested under e and has minimum u'. If $u' = u$, that is e and e' are two consecutive edges of P, then let R be the region of L that has u as the right endpoint. We delete e, add a vertex w in the region R (and so we have $w \prec u \prec v$). See Figure 8 (left) for this replacement. If $u' \neq u$, then let R be the region (u, u') of L. We delete e, add a vertex w in the region R (and so we have $u \prec w \prec u' \prec v$). See Figure 8 (right) for this replacement. For every of these cases, we set $\sigma(uw) = \sigma(wv) = \sigma(e)$. □

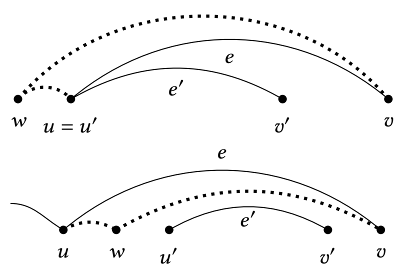

Fig. 8. The subdivision of the edge $e = uv$ when u is a vertex of degree 2 while containing the edge $e' = u'v'$ by Lemma 51: when e' is incident to u (left) and when e' is not incident to u (right).

By exhaustively applying Lemma 51, we can transform paths of length at least two into arbitrarily long proper paths while preserving yes-instances. To obtain our kernel, we will however need to shorten sufficiently long paths while preserving yes-instances. The following lemma allows us to handle this in case where all the considered paths are sufficiently long.

Lemma 52. *Let $k \geq 3$, G be a graph and $\mathcal{P}$ be a set of paths each of length more than $(|V(G)| + 1)2^{|\mathcal{P}|}|\mathcal{P}|$ such that G and $\mathcal{P}$ are near-disjoint. Then, $G \vee \mathcal{P}$ admits a k-page book embedding if and only if there exists a set of paths $\mathcal{P}'$ each of length at most $(|V(G)| + 1)2^{|\mathcal{P}|}|\mathcal{P}|$ that serves as $\mathcal{P}$ such that $G \vee \mathcal{P}'$ also admits a k-page book embedding.*

Proof. We start with the forward direction. Consider a k-page book embedding L for $G \vee \mathcal{P}$ and let L_G be the restriction of L to G. First we establish that, given any region R of L_G, in L we can replace $\mathcal{P}$ with a set of paths $\mathcal{P}'$ that serves as $\mathcal{P}$ such that every path of $\mathcal{P}'$ touches the region R most $2^{|\mathcal{P}|}|\mathcal{P}|$ times.

If every path of $\mathcal{P}$ touches R at most $2^{|\mathcal{P}|}|\mathcal{P}|$ times, then the statement is true for $\mathcal{P}' = \mathcal{P}$. Suppose otherwise, and let $\mathcal{R}$ be the set of all paths in $\mathcal{P}$ that touch the region R.

For every $P \in \mathcal{R}$, say P has vertex set $\{u_1, \dots, u_t\}$ for some $t \geq 2$ with edges $u_i u_{i+1}$ for every $i \in [t-1]$, and let Z_P be the set of all vertices of P in the region R that have either minimum or maximum label. Clearly, since $P \in \mathcal{R}$, the set Z_P must have either one or two elements. If $Z_P = \{a\}$, we define u_P and v_P both to be equal to a. If $Z_P = \{a, b\}$ with $a \prec b$, we define $u_P = a$ and $z_P = b$.

The idea now is to delete the subpath P_{uv} of P between u_P and v_P and create a path S_P such that S_P **(1)** has bounded length, **(2)** serves as P_{uv} and **(3)** is completely contained in R. Intuitively, the length of the path S_P will be at most the number of vertices between u_P and v_P: we will show that we will always be able to "jump" at least one vertex with an edge. For the argument to work, the replacement of the path segments P_{uv} with S_P will be carried out using a recursive strategy where the next path to be replaced is selected based on a specific condition.

We initiate by having $\mathcal{R}$ contain all the paths that touch region R, as introduced earlier. Moreover, we set $\mathcal{R}' = \emptyset$ to be the empty set, and we will use $\mathcal{R}'$ to store paths which have already been processed. Inductively, until $\mathcal{R} = \emptyset$, we consider a path $P \in \mathcal{R}$ such that for every other path $P' \in \mathcal{R}$ we have $(u_{P'}, v_{P'}) \nsubseteq (u_P, v_P)$, *i.e.*, the interval between $u_{P'}$ and $v_{P'}$ does not fully contain the corresponding interval for any other path in $\mathcal{R}$. Let M_P be the set defined as follows

$$M_P = \Big(\bigcup_{P'' \in \mathcal{R}} \{u_{P''}, v_{P''}\} \cup \bigcup_{P' \in \mathcal{R}'} V(P')\Big) \cap (u_P, v_P)$$

The set M_P represents the set of all vertices, that are present at this stage, in the interval (u_P, v_P).

If $M_P = \emptyset$ then we either do not do anything if $u_P = v_P$ or add the edge $u_P v_P$ and set $\sigma(u_P v_P) = 1$. Suppose $M_P \neq \emptyset$ and let $M_P = \{v_0, \dots, v_t\}$ for some $t \geq 0$ and assume $v_0 \prec \dots \prec v_t$. We add a path S_P in the following way: if $t = 0$, we add the edge $u_P v_P$ and set $\sigma(u_P v_P) = \min([h] \setminus \sigma(v_0))$. If $t \geq 1$, S_P has t internal vertices $\{u_1, \dots, u_t\}$ such that $v_0 \prec u_1 \prec v_1 \prec \dots \prec u_t \prec v_t$. We add the edges $u_P u_1$, $u_t v_P$ and $u_i u_{i+1}$, for every $i \in [t-1]$. Finally we set $\sigma(u_P u_1) = \min([h] \setminus \sigma(v_0))$, $\sigma(u_t v_P) = \min([h] \setminus \sigma(v_t))$ and $\sigma(u_i u_{i+1}) = \min([h] \setminus \sigma(v_i))$ for every $i \in [t-1]$.

Now we show that the way we assigned pages to the edges of S_P do not any create edge crossings. Since $h \geq 3$ and every vertex in M_P has at most two incident edges (thus resulting in $|\sigma(v_i)| \leq 2$), the sets $[h] \setminus \sigma(v_0)$, $[h] \setminus \sigma(v_t)$ and $[h] \setminus \sigma(v_i)$ for every $i \in [t-1]$ are not empty: since the edge of S_P used to *jump* v_i can not be assigned to any of the pages in $\sigma(v_i)$, we can always assign a page in $[h]$ to every edge of S_P in such a way no edge crossing is created. We can finally define P' as the path obtained from P by replacing P_{uv} with S_P.

Now we are left to show how many vertices are necessary to construct S_P at each step: recall that this number is equal to $|M_P| - 1$. Let i be an integer such that $0 \leq i \leq |\mathcal{R}| \leq |\mathcal{P}|$. We aim to evaluate $T(i+1)$, that is the maximum number of internal vertices of the $(i+1)$-th subpath S_P. Let $N_P = (\bigcup_{P'' \in \mathcal{R}} \{u_{P''}, v_{P''}\}) \cap (u_P, v_P)$ and $O_P = (\bigcup_{P' \in \mathcal{R}'} V(P')) \cap (u_P, v_P)$ and note that $M_P = N_P \cup O_P$.

Let us first analyze the case $i = 0$, that is $T(1)$. Since $\mathcal{R}'$ is empty, S_P creates $|N_P| - 1$ new vertices and so we have $T(1) = |N_P| - 1 \leq |\mathcal{R}| \leq |\mathcal{P}|$.

Let us consider the case $i \geq 1$. The set N_P contains at most one vertex per path of $\mathcal{R}$: this is ensured by the choice of P. For this reason, we have $|N_P| \leq |\mathcal{R}| \leq |\mathcal{P}|$. The set O_P might contain every vertex in R, both endpoint and internal,

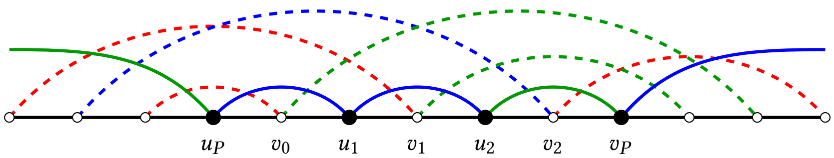

Fig. 9. An example where black vertices and full edges are part of P', the path obtained from P by replacing the subpath P_{uv} between u_P and v_P with S_P. Note that edges with different colors belong to different pages of the book embedding.

of every path of $\mathcal{R}'$. For this reason we have $|O_P| \leq 2|\mathcal{R}'| + \sum_{j=1}^{i} T(j)$. Now we have that

$$\begin{aligned} T(i+1) &\leq |\mathcal{R}| + 2|\mathcal{R}'| + \textstyle\sum_{j=1}^{i} T(j) \\ &\leq 2|\mathcal{P}| + \textstyle\sum_{j=1}^{i} T(j) \\ &\leq T(i) + 2|\mathcal{P}| + \textstyle\sum_{j=1}^{i-1} T(j) \\ &\leq 2T(i) \\ &\leq 2^i T(1) \\ &\leq 2^i |\mathcal{P}| \end{aligned}$$

To summarize we have found out that $T(i+1) \leq 2^i|\mathcal{P}|$ for every $0 \leq i \leq |\mathcal{R}| \leq |\mathcal{P}|$. In particular, we have found a set $\mathcal{P}'$ that serves as $\mathcal{P}$, where every path in $\mathcal{P}'$ touches the region R at most $2^{|\mathcal{P}|}|\mathcal{P}|$ times. At this point we set $\mathcal{R} := \mathcal{R} \setminus \{P\}$ and $\mathcal{R}' := \mathcal{R}' \cup \{P'\}$ and, if possible, select another path of $\mathcal{R}$.

Note that by the construction described in the paragraphs above, applying this procedure to a region R does not increase the number of times paths of $\mathcal{P}$ touch regions that are not R. This means we can apply this procedure on one region at a time until the claim holds for every region and thus satisfying the statement. This provides a k-page book embedding of $G \vee \mathcal{P}'$ with the desired property.

Let us consider the backwards direction. Suppose there exists a set $\mathcal{P}'$ that serve as $\mathcal{P}$, where every path in $\mathcal{P}'$ has length at most $(|V(G)|+1)2^{|\mathcal{P}|}|\mathcal{P}|$ and $G \vee \mathcal{P}'$ admits a k-page book embedding. Since each path of $\mathcal{P}$ is of length more than $(|V(G)|+1)2^{|\mathcal{P}|}|\mathcal{P}|$, we apply Lemma 51 the appropriate number of times on the paths of $\mathcal{P}'$ so that the lengths of the paths coincide with the ones of $\mathcal{P}$ and obtain that $G \vee \mathcal{P}$ admits a k-page book embedding. □

The following result is a direct consequence of combining Lemma 51 and 52.

Corollary 53. *Let $k \geq 3$, G be a graph and $\mathcal{P}$ be a set of paths each of length more than $(|V(G)|+1)2^{|\mathcal{P}|}|\mathcal{P}|$ such that G and $\mathcal{P}$ are near-disjoint. Then, $G \vee \mathcal{P}$ admits a k-page book embedding if and only if $G \vee \mathcal{P}'$ admits a k-page book embedding where $\mathcal{P}'$ is a set of paths each of length exactly $(|V(G)|+1)2^{|\mathcal{P}|}|\mathcal{P}|$ that serves as $\mathcal{P}$.*

Proof. Let us start with the forward direction. Suppose that $G \vee \mathcal{P}$ admits a k-page book embedding. We apply the forward direction of Lemma 52 and that $G \vee \mathcal{P}'$ admits a k-page book embedding where every path of $\mathcal{P}'$ has length at most $(|V(G)|+1)2^{|\mathcal{P}|}|\mathcal{P}|$. Now by Lemma 51, we obtain obtain that $G \vee \mathcal{P}''$ admits a k-page book embedding where every path of $\mathcal{P}''$ has length exactly $(|V(G)|+1)2^{|\mathcal{P}|}|\mathcal{P}|$.

The reverse direction follows directly from Lemma 51. □

Putting Everything Together. At this point, we have all the ingredients needed to establish the fixed-parameter tractability of Book Thickness with respect to the feedback edge number.

Proof of Theorem 46. Let (G, k) be an input of Book Thickness. We establish fixed-parameter tractability by constructing a problem kernel in polynomial time. By Observation 50, we compute $G_{\geq 2}$, F, G_F and B_F. We set $B := B_F$ and let $\mathcal{P} := \mathcal{P}_F$ be the set all maximal proper paths of G_F for B. Note that, given B, the set $\mathcal{P}_F$ can be computed using Observation 50. Let $\mathcal{P}_{\leq}$ be the set of all the paths of $\mathcal{P}$ of length at most $(|B| + 1)2^{|\mathcal{P}|}|\mathcal{P}|$.

At this point we start a loop that repeats at most $|\mathcal{P}| \leq 4$fen times: at each iteration either we obtain a kernel of the desired size and the algorithm ends, or the size of $\mathcal{P}$ is reduced by at least one. If $\mathcal{P}_{\leq} = \emptyset$ holds, that is, all the paths in $\mathcal{P}$ have length more than $(|B|+1)2^{|\mathcal{P}|}|\mathcal{P}|$, Lemma 52 can be applied to obtain a kernel of the desired size and the algorithm ends. If $\mathcal{P}_{\leq} \neq \emptyset$ holds; denote with $V_{\leq}$ the set of vertices in paths of $\mathcal{P}_{\leq}$. Note that $|V_{\leq}| \leq (|B| + 1)2^{|\mathcal{P}|}|\mathcal{P}||\mathcal{P}_{\leq}| \leq (|B| + 1)2^{|\mathcal{P}|}|\mathcal{P}|^2$. In this case, We update the sets B and $\mathcal{P}$ by setting the former to $B \cup V_{\leq}$ and the latter to $\mathcal{P} \setminus \mathcal{P}_{\leq}$, and enter the next iteration of the loop.

To conclude the proof, it suffices to provide an upper bound to the size of a graph created in this way. The largest graph obtained by this construction results from there being $|\mathcal{P}|$ iterations, whereas in each iteration there is only a single path in $\mathcal{P}_{\leq}$ and this path is of maximum length. Let $S(i + 1)$ be an upper bound on the number of vertices that have been added to B after the i-th step of the recursion. For $i = 0$, we have that $S(1) = |B_F|$.

Let us consider the case $i \geq 1$. Together with the vertices that were present after the $(i-1)$-th step, that is $S(i)$, we also have to consider the vertices of a unique path having maximum length allowed at this step, that is $2^{|\mathcal{P}|-i}(|\mathcal{P}|-i)(S(i)+1)$. Now we have that:

$$\begin{aligned} S(i+1) &\leq S(i) + 2^{|\mathcal{P}|-i}|(\mathcal{P}| - i)(S(i) + 1) \\ &\leq S(i) + 2^{|\mathcal{P}|-i}(|\mathcal{P}| - i)S(i) + 2^{|\mathcal{P}|-i}(|\mathcal{P}| - i) \\ &\leq 3 * 2^{|\mathcal{P}|-i}(|\mathcal{P}| - i)S(i) \end{aligned}$$

Hence, the total size of the obtained kernel can be upper-bounded by $(3 * 2^{|\mathcal{P}|-i}(|\mathcal{P}| - i))^{|\mathcal{P}|} \cdot S(1)$, which is at most $2^{O(\mathsf{fen}(G)^2)}$. □

6.2 An FPT-algorithm for SUBHAM using the Feedback Edge Number

In this section, we provide a linear kernel for SUBHAM parameterized by the feedback edge number. The main idea is to reduce the size of the tree $G - F$, where F is minimum feedback edge set of the input graph G. To do so we need the following simple corollary and lemma that allow us to bound the number of leaves and vertices of degree at most two in G. We have already seen in Lemma 47 that we can remove vertices of degree at most one. The next lemma allows us to bound the number of vertices of degree two.

Lemma 54. *Let G be a graph and P be a path of length 4 in G such that all inner vertices of P have degree two. Let G' be a graph obtained from G by contracting any edge on P. Then, G is subhamiltonian if and only if so is G'.*

Proof. Let's assume that G' is subhamiltonian with witness (G_H, H) and $P' = [v_b, v_1, v_2, v_e]$ is a path P after contracting. Due to the fact that $deg_G(v_1) = deg_G(v_2) = 2$ and $2 \leq deg_{G_H}(v_1), deg_{G_H}(v_2) \leq 4$, there exist a face f_H in drawing G'_H such that $(v_1, v_2) \in E(f_H)$ and $E(f_H) \cap E(H) \neq \emptyset$. From Observation 4 *(2)* and *(1)* we obtain that G is subhamiltonian.

Let's assume that G is subhamiltonian with witness (G_H, H). Let P be a path $[v_b, v_1, v_2, v_3, v_e]$ and D be a drawing of G that respects H. All inner vertices from $V(P)$ have degree two, which implies that there exists a face f of D such that $V(P) \subseteq V(f)$. From Lemma 14 applied to $v_b v_e$ and a face f, we get new witness $(G_{H'}, H')$ and H' crosses $v_b v_e$ at most in two points. There cannot be three edges which have one ending in v_1, v_2 and v_3 and crosses $v_b v_e$, so at least one pair of varieties (v_b, v_2), (v_1, v_3) or (v_2, v_e) are in the same face together, so from Observation 2 we can connect them. From Observation 4 *(1)* we obtain that G' is subhamiltonian. □

We are now ready to provide our kernel for SUBHAM.

Theorem 55. SUBHAM *parameterized by the feedback edge number* k *admits a kernel with at most* $12k - 8$ *vertices and at most* $14k - 9$ *edges.*

Proof. Let G be a connected graph, *i.e.*, the given instance of SUBHAM. We first use Lemma 47 to ensure that G has no leaves. We now compute a minimum feedback edge set $F \subseteq E(G)$ for G using Fact 1. Let T be the tree $G - F$. Note that T has at most $2|F|$ many leaves, since every leaf of T must be adjacent to an edge in F. This also implies that T has at most $|L| - 2$ vertices of degree at least 3, where L is the set of all leaves of T. Therefore, it only remains to obtain an upper bound on the vertices having degree exactly two in T. We say that a path P in T is *proper* if it is an inclusion-wise maximal path in T having only inner vertices of degree two in G. Because of Lemma 54, we can assume that any proper path in T has length at most three. Also note that every vertex having degree two in T is an inner vertex of such a maximal path. Moreover, since every proper path must have both of its endpoints in $V(F) \cup B$, where B is the set of all vertices having degree at least three in T, the number of distinct proper paths in T is equal to the number of edges in a tree with $|V(F) \cup B|$ many vertices. Therefore, the number of proper path in T is at most $|V(F) \cup B| - 1 \leq 2|F| + 2|F| - 2 - 1 = 4|F| - 3$. Since every proper path contains at most two vertices of degree two in T, we obtain that T contains at most $2(4|F| - 3) = 8|F| - 6$ vertices of degree two. Altogether, T contains at most $2|F| + 2|F| - 2 + 8|F| - 6 = 12|F| - 8$ vertices and at most $12|F| - 9$ edges. Therefore, G has at most $12|F| - 8$ vertices and at most $14|F| - 9$ edges. Finally, the time required to obtain the kernel is at most $O(|V(G)| + |E(G)|)$. □

Theorem 45 now follows directly from Theorems 55 and 46. Moreover, by combining Theorem 55 with the subexponential algorithm of Corollary 21, we can slightly strengthen our main result as follows.

Corollary 56. SUBHAM *can be solved in time* $2^{O(\sqrt{k})} \cdot n^{O(1)}$, *where* k *is the feedback edge number of the input graph.*

7 Concluding Remarks

While our main algorithmic result settles the complexity of computing 2-page book embeddings under the exponential time hypothesis, many questions remain when one aims at computing k-page book embeddings for a fixed k greater than 2. To the best of our knowledge, even the existence of a single-exponential algorithm for this problem is open.

In terms of the problem's parameterized complexity, it is natural to ask whether one can obtain a generalization of Theorem 20 for computing k-page book embeddings when $k > 2$. In fact, it is entirely open whether computing, e.g., 4-page book embeddings is even in XP when parameterized by the treewidth. In this sense, our positive result for the feedback edge number can be seen as a natural step on the way towards finally settling the structural boundaries of tractability for computing page-optimal book embeddings.

Acknowledgments

Robert Ganian was supported by Project No. Y1329 of the Austrian Science Fund (FWF) and Project No. ICT22-029 of the Vienna Science Foundation (WWTF). Sebastian Ordyniak was supported by the Engineering and Physical Sciences Research Council (EPSRC) (Project EP/V00252X/1).

References

[1] Bernardo M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Pedro Ramos, and Gelasio Salazar. The 2-page crossing number of K_n. *Discrete & Computational Geometry*, 49(4):747–777, 2013. doi:10.1007/S00454-013-9514-0.

[2] Patrizio Angelini, Marco Di Bartolomeo, and Giuseppe Di Battista. Implementing a partitioned 2-page book embedding testing algorithm. *Proc. GD 2012*, 7704:79–89, 2012. doi:10.1007/978-3-642-36763-2_8.

[3] Michael J. Bannister and David Eppstein. Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. *Journal of Graph Algorithms and Applications*, 22(4):577–606, 2018. doi:10.7155/jgaa.00479.

[4] Giuseppe Di Battista and Roberto Tamassia. Incremental planarity testing. *Proc. FOCS 1989*, pages 436–441, 1989. doi:10.1109/SFCS.1989.63515.

[5] Michael A. Bekos, Martin Gronemann, and Chrysanthi N. Raftopoulou. Two-page book embeddings of 4-planar graphs. *Algorithmica*, 75(1):158–185, 2016. doi:10.1007/s00453-015-0016-8.

[6] Frank Bernhart and Paul C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979. doi:10.1016/0095-8956(79)90021-2.

[7] Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. *Journal of Graph Algorithms and Applications*, 24(4):603–620, 2020. doi:10.7155/jgaa.00526.

[8] Daniel Bienstock and Clyde L. Monma. Optimal enclosing regions in planar graphs. *Networks*, 19(1):79–94, 1989. doi:10.1002/NET.3230190107.

[9] Daniel Bienstock and Clyde L. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5(1):93–109, 1990. doi:10.1007/BF01840379.

[10] Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015. doi:10.1016/J.IC.2014.12.008.

[11] F. Chung, F. Leighton, and A. Rosenberg. Embedding graphs in books: a layout problem with applications to VLSI design. *SIAM Journal on Algebraic Discrete Methods*, 8(1):33–58, 1987. doi:10.1137/0608002.

[12] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

[13] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

[14] Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *Journal of the ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.

[15] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Transactions on Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707.

[16] Hubert de Fraysseix, Patrice Ossona de Mendez, and János Pach. A left-first search algorithm for planar graphs. *Discrete & Computational Geometry*, 13:459–468, 1995. doi:10.1007/BF02574056.

[17] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[18] Frederic Dorn, Eelko Penninkx, Hans L. Bodlaender, and Fedor V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica*, 58(3):790–810, 2010. doi:10.1007/S00453-009-9296-1.

[19] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

[20] Vida Dujmović and David R. Wood. On linear layouts of graphs. *Discrete Mathematics & Theoretical Computer Science*, 6(2):339–358, 2004. doi:10.46298/dmtcs.317.

[21] Vida Dujmovic and David R. Wood. Graph treewidth and geometric thickness parameters. *Discrete & Computational Geometry*, 37(4):641–670, 2007. doi:10.1007/s00454-007-1318-7.

[22] Toshiki Endo. The pagenumber of toroidal graphs is at most seven. *Discrete Mathematics*, 175(1):87–96, 1997. doi:10.1016/S0012-365X(96)00144-6.

[23] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.

[24] Robert Ganian, Fabrizio Montecchiani, Martin Nöllenburg, and Meirav Zehavi. Parameterized complexity in graph drawing (dagstuhl seminar 21293). *Dagstuhl Reports*, 11(6):82–123, 2021. doi:10.1016/j.artint.2017.12.006.

[25] Joseph L. Ganley and Lenwood S. Heath. The pagenumber of k-trees is $O(k)$. *Discrete Applied Mathematics*, 109(3):215–221, 2001. doi:10.1016/S0166-218X(00)00178-5.

[26] M. R. Garey, D. S. Johnson, and R. Endre Tarjan. The planar hamiltonian circuit problem is np-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. doi:10.1137/0205049.

[27] Emilio Di Giacomo and Giuseppe Liotta. The hamiltonian augmentation problem and its applications to graph drawing. *Proc. WALCOM 2010, LNCS*, 5942:35–46, 2010. doi:10.1007/978-3-642-11440-3_4.

[28] Qian-Ping Gu and Hisao Tamaki. Improved bounds on the planar branchwidth with respect to the largest grid minor size. *Algorithmica*, 64(3):416–453, 2012. doi:10.1007/S00453-012-9627-5.

[29] Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. *Proc. GD 2000*, 1984:77–90, 2000. doi:10.1007/3-540-44541-2_8.

[30] Carsten Gutwenger, Petra Mutzel, and René Weiskircher. Inserting an edge into a planar graph. *Algorithmica*, 41(4):289–308, 2005. doi:10.1007/S00453-004-1128-8.

[31] András Gyárfás and Jenö Lehel. Covering and coloring problems for relatives of intervals. *Discrete Mathematics*, 55(2):167–180, 1985. doi:10.1016/0012-365X(85)90045-7.

[32] Christian Haslinger and Peter F. Stadler. RNA structures with pseudo-knots: Graph-theoretical, combinatorial, and statistical properties. *Bulletin of Mathematical Biology*, 61(3):437–467, 1999. doi:10.1006/bulm.1998.0085.

[33] Lenwood S. Heath. Embedding outerplanar graphs in small books. *SIAM Journal on Algebraic Discrete Methods*, 8(2):198–218, 1987. doi:10.1137/0608018.

[34] Michael Hoffmann and Boris Klemz. Triconnected planar graphs of maximum degree five are subhamiltonian. *Proc. ESA 2019, LIPIcs*, 144(58):1–14, 2019. doi:10.4230/LIPIcs.ESA.2019.58.

[35] Seok-Hee Hong and Hiroshi Nagamochi. Two-page book embedding and clustered graph planarity. Technical report, Citeseer, 2009.

[36] Seok-Hee Hong and Hiroshi Nagamochi. Simpler algorithms for testing two-page book embedding of partitioned graphs. *Theoretical Computer Science*, 725:79–98, 2018. doi:10.1016/J.TCS.2015.12.039.

[37] John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974. doi:10.1145/321850.321852.

[38] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/JCSS.2001.1774.

[39] Hugo Jacob and Marcin Pilipczuk. Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs. *Proc. WG 2022*, 13453:287–299, 2022. doi:10.1007/978-3-031-15914-5_21.

[40] Tuukka Korhonen. A single-exponential time 2-approximation algorithm for treewidth. *Proc. FOCS 2021*, pages 184–192, 2021. doi:10.1109/FOCS52979.2021.00026.

[41] Germain Kreweras. Sur les partitions noncroisees d'un cycle. *Discrete Mathematics*, 1, 1972.

[42] Seth M. Malitz. Genus g graphs have pagenumber $O(\sqrt{g})$. *Journal of Algorithms*, 17(1):85–109, 1994. doi:10.1006/jagm.1994.1028.

[43] Dániel Marx. Four shorts stories on surprising algorithmic uses of treewidth. *Treewidth, Kernels, and Algorithms*, 12160:129–144, 2020. doi:10.1007/978-3-030-42071-0_10.

[44] Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. A subexponential parameterized algorithm for directed subset traveling salesman problem on planar graphs. *SIAM Journal on Computing*, 51(2):254–289, 2022. doi:10.1137/19M1304088.

[45] Petra Mutzel. The SPQR-tree data structure in graph drawing. *Proc. ICALP 2003*, 2719:34–46, 2003. doi:10.1007/3-540-45061-0_4.

[46] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

[47] Malgorzata Nowicka, Vinay K. Gautam, and Pekka Orponen. Automated rendering of multi-stranded dna complexes with pseudoknots, *Proc. UCNC 2024*, 14776:190–202, 2023. https://doi.org/10.1007/978-3-031-63742-1_14.

[48] Neil Robertson and Paul D. Seymour. Graph minors. x. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991. doi:10.1016/0095-8956(91)90061-N.

[49] Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, 1994. doi:10.1006/JCTB.1994.1073.

[50] Rodica Simion. Noncrossing partitions. *Discrete Mathematics*, 217(1):367–409, 2000. doi:10.1016/S0012-365X(99)00273-3.

[51] Johannes Uhlmann and Mathias Weller. Two-layer planarization parameterized by feedback edge set. *Theoretical Computer Science*, 494:99–111, 2013. doi:10.1016/J.TCS.2013.01.029.

[52] Avi Wigderson. The complexity of the hamiltonian circuit problem for maximal planar graphs. *Technical Report*, 1982. doi:10.1137/0205049.

[53] Mihalis Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36–67, 1989. doi:10.1016/0022-0000(89)90032-9.