



Recognizing Map Graphs of Bounded Treewidth

Patrizio Angelini¹ · Michael A. Bekos² · Giordano Da Lozzo³ ·
Martin Gronemann⁴ · Fabrizio Montecchiani⁵ · Alessandra Tappini⁵

Received: 3 July 2022 / Accepted: 9 October 2023 / Published online: 27 October 2023
© The Author(s) 2023

Abstract

A map is a partition of the sphere into interior-disjoint regions homeomorphic to closed disks. Some regions are labeled as nations, while the remaining ones are labeled as holes. A map in which at most k nations touch at the same point is a k -map, while it is hole-free if it contains no holes. A graph is a map graph if there is a bijection between its vertices and the nations of a map, such that two nations touch if and only if the corresponding vertices are connected by an edge. We present a fixed-parameter tractable algorithm for recognizing map graphs parameterized by treewidth. Its time complexity is linear in the size of the graph. It reports a certificate in the form of a so-called witness, if the input is a yes-instance. Our algorithmic framework is general enough to test, for any k , if the input graph admits a k -map or a hole-free k -map.

Keywords Map graphs · K-map graphs · Fixed-parameter tractability · Treewidth

1 Introduction

Planarity is one of the most influential concepts in Graph Theory. Inspired by topological inference problems and by intersection graphs of planar curves, in 1998, Chen, Grigni and Papadimitriou [2] suggested the study of map graphs as a generalized notion of planarity. A *map* of a graph G is a function \mathcal{M} that assigns each vertex v of G to a region $\mathcal{M}(v)$ on the sphere homeomorphic to a closed disk such that no two regions share an interior point, and any two distinct vertices v and w are adjacent in

Patrizio Angelini, Michael A. Bekos, Giordano Da Lozzo, Martin Gronemann, Fabrizio Montecchiani and Alessandra Tappini have contributed equally to this work.

An extended abstract of this manuscript has appeared in the proceedings of the 18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022) [1].

✉ Fabrizio Montecchiani
fabrizio.montecchiani@unipg.it

Extended author information available on the last page of the article

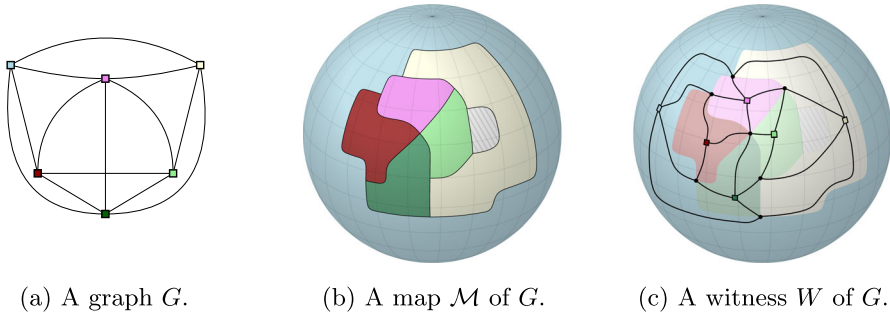


Fig. 1 **a** A graph G , **b** a map of G - the striped region is a hole, and **c** a witness of G (Color figure online)

G if and only if the boundaries of $\mathcal{M}(v)$ and $\mathcal{M}(w)$ share at least one point. For each vertex v of G , the region $\mathcal{M}(v)$ is called the *nation* of v . A connected open region of the sphere that is not covered by nations is a *hole*. A graph that admits a map is a *map graph*, whereas a graph that admits a map without holes is a *hole-free map graph*. Moreover, a graph that admits a map in which at most k nations meet at a common point is a *k -map graph*. Observe that, when $k \geq n - 1$, map graphs and k -map graphs trivially coincide. Fig 1a and b show a graph and a map of it, respectively. Map graphs generalize planar graphs by allowing local non-planarity at points where more than three nations meet. In fact, the planar graphs are exactly those graphs having a 3-map [2, 3].

Besides their theoretical interest, the study of map graphs is motivated by applications in graph drawing, circuit board design, and topological inference problems [4–7]. Map graphs are also useful to design parameterized and approximation algorithms for several optimization problems that are NP-hard on general graphs [8–12].

A natural and central algorithmic question concerns the existence of efficient algorithms for recognizing map graphs. Towards an answer to this question, Chen et al. [2, 13] first gave a purely combinatorial characterization of map graphs: A graph is a map graph if and only if it admits a witness, formally defined as follows; see Fig 1c. A *witness* of a graph $G = (V, E)$ is a bipartite planar graph $W = (V \cup I, A)$ with $A \subseteq V \times I$ and such that $W^2[V] = G$, where the graph $W^2[V]$ is the *half-square* of W , that is, the graph on the vertex set V in which two vertices are adjacent if and only if their distance in W is two. Here, the vertices in I are meant to represent the adjacencies among nations, hence they are called *intersection vertices*, while the vertices in V are called *real*. Note that the degree of an intersection vertex corresponds to the number of nations that meet at the corresponding point in the map. Since W can always be chosen to have linear size in the number of vertices of G [13], the problem of recognizing map graphs is in NP. In 1998, Thorup [14] proposed a polynomial-time algorithm to recognize map graphs. However, the extended abstract by Thorup does not contain a complete proof of the result and, to the best of our knowledge, a full version has not appeared yet. Moreover, the proposed algorithm has two drawbacks. First, the time complexity is not specified explicitly (the exponent of the polynomial bounding the time complexity is estimated to be about 120 [15]; see also [6, 16]).

Second, it does not report a certificate in the positive case; a natural one would be a witness.

Hence, the problem of finding a simple and efficient recognition algorithm for map graphs remains open. In recent years, several authors focused on graphs admitting restricted types of maps. For instance, Chen studied the density of k -map graphs [17]. As another example, in a recent milestone paper on linear layouts, Dujmović et al. [18] proved that the queue number of k -map graphs is cubic in k ; this bound has been recently improved to linear [19]. On a similar note, Brandenburg [20] showed that k -map graphs admit book embeddings in $6\lfloor \frac{k}{2} \rfloor + 5$ pages. Chen et al. [15] focused on hole-free 4-map graphs and gave a cubic-time recognition algorithm for this graph family. Later, Brandenburg [6] gave a cubic-time recognition algorithm for general (i.e., not necessarily hole-free) 4-map graphs, by exploiting an alternative characterization of these graphs closely related to maximal 1-planarity. Notably, a polynomial-time recognition algorithm for the family of (general or hole-free) k -map graphs with $k > 4$ is still missing. In particular, for $k > 4$, the only result we are aware of is a characterization of 5-map graphs in terms of crossing patterns [5]. A different approach for the original problem is the one by Mních, Rutter, and Schmidt [16], who proposed a linear-time algorithm to recognize the map graphs with an outerplanar witness; a certificate witness, if any, is also reported. In addition, we mention the series of works of Le and Le [21–23]; among other results, they proved that, for any fixed $g \geq 8$, deciding if a given n -vertex m -edge graph G admits a map having a witness of girth at least g can be done in $O(n^2m)$ time [21].

We remark that the size of the largest clique in a k -map graph is $\lfloor 3k/2 \rfloor$ (see, e.g., [13]), thus bounding the size of the largest clique does not seem to be a strong enough structural limitation of the input to obtain an efficient time complexity. In particular, map graphs have $O(n)$ maximal cliques [13], and such cliques can be identified in $O(n^4)$ time by looking for maximal independent sets in the complement graph [24]. Despite the notable amount of work, no prior research focuses on further structural parameters of the input graph to design efficient recognition algorithms. In this paper, we address precisely this challenge.

Our contribution Our main result is a novel algorithmic framework that can be used to recognize map graphs, as well as variants thereof; in particular, hole-free k -map graphs and k -map graphs. Recall that, by setting $k = n - 1$, our algorithm also recognizes (hole-free) map graphs. In fact, we can also compute the minimum value of k within the same asymptotic running time. The proposed algorithm is parameterized by the treewidth [25, 26] of the n -vertex input graph G and its time complexity has a linear dependency in n , while it does not depend on the natural parameter k . Notably, for graphs of bounded treewidth, our algorithm improves over the existing literature [6, 14, 15] in three ways: it solves the problem for *any* fixed k , it can deal with *both* scenarios where holes are or are not allowed in the sought map (which is the case also for the result in [6]), and it exhibits a linear running time. The following theorem summarizes our main contribution.

Theorem 1 *Given an n -vertex graph G and a tree-decomposition of G of width t , there is a $O(t^{O(t)} \cdot n)$ -time algorithm that computes the minimum k , if any, such that*

G admits a (hole-free) k -map. In the positive case, the algorithm returns a certificate in the form of a witness of G within the same time complexity.

We remark that the problem of recognizing map graphs can be expressed by using MSO_2 logic. Thus the main positive result behind Theorem 1 can be alternatively achieved by Courcelle’s theorem [27]. A formal proof is reported in the appendix. However, with this approach, the dependency of the time complexity on the treewidth is notoriously very high. As a matter of fact, Courcelle’s theorem is generally used as a classification tool, while the design of an explicit ad-hoc algorithm remains a challenging and valuable task [28].

To prove Theorem 1, we first solve the decision version of the problem. For a fixed k , we use a dynamic-programming approach, which can deal with different constraints on the desired witness. While we exploit such flexibility to check whether at most k nations intersect at any point and whether holes can be avoided, other constraints could be plugged into the framework such as, for example, the outerplanarity of the witness (as in [16]). In view of this versatility, future applications of our tools may be expected.

Proof strategy We exploit the characterization in [13] and test for the existence of a suitable witness of the input graph. The crux of our technique is in the computation of suitable records that represent equivalent witnesses and contain only vertices of a tree-decomposition bag. Each such record must carry enough information, in terms of an embedding of the witness, so to allow testing whether it can be extended with a new vertex or merged with another witness. Moreover, we need to check whether any such witness yields a k -map and, if required, a hole-free one. To deal with the latter property, we provide a strengthening of the characterization in [13], which we believe to be of independent interest, that translates into maintaining suitable counters on the edges of our records. Additional checks on the desired witness can be plugged in the presented algorithmic framework, provided that the records store enough information. One of the main difficulties is hence “sketching” irrelevant parts of the embedded graph without sacrificing too much information. (A similar challenge is faced in the context of different planarity and beyond-planarity problems [29–31].) Also, when creating such sketches, multiple copies (potentially linearly many) of the same edge may appear, which we need to simplify in order to keep our records small. The formalization of such records then allows us to exploit a dynamic-programming approach on a tree-decomposition.

Paper structure Section 2 contains preliminary definitions. Section 3 illustrates basic properties of map graphs that will be used throughout the paper. Section 4 introduces the concept of “sketching” an embedding of a witness, the key ingredient of the algorithmic framework, which we present in Sect. 5. Section 6 contains open problems raised by our work.

2 Preliminaries

We consider finite, undirected, and simple graphs, although some procedures may produce non-simple graphs. In such a case the presence of self-loops or multiple

edges will be clearly indicated. Let $G = (V, E)$ be a graph; for a vertex $v \in V$, we denote by $N(v)$ the set of neighbors of v in G , and by $\text{deg}(v)$ the *degree* of v , i.e., the cardinality of $N(v)$.

Embeddings A *topological embedding* of a graph G on the sphere Σ is a representation of G on Σ in which each vertex of G is associated with a point and each edge of G with a simple arc between its two endpoints in such a way that any two arcs intersect only at common endpoints. A topological embedding of G subdivides the sphere into topologically connected regions, called *faces*. If G is connected, the *boundary* of a face f is a closed walk, that is, a circular list of alternating vertices and edges; otherwise, the boundary of f is a *set* of closed walks. Note that a cut-vertex of G may appear multiple times in any such walk. A topological embedding of G uniquely defines a *rotation system*, that is, a cyclic order of the edges around each vertex. If G is connected, the boundary defining each face can be reconstructed from a rotation system; otherwise, to reconstruct the boundary of every face f , we also need to know which connected components are incident to f . We call the incidence relationship between closed walks of different components and faces the *position system* of G . A *combinatorial embedding* of G is an equivalence class of topological embeddings that define the same rotation and position systems. An *embedded graph* G is a graph along with a combinatorial embedding. A pair of parallel edges e and e' of G with end-vertices v and w is *homotopic* if there is a face of G whose boundary consists of a single closed walk $\langle v, e, w, e' \rangle$.

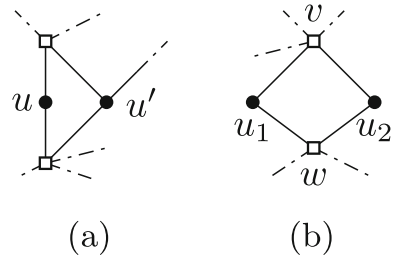
Tree-decompositions Let (\mathcal{X}, T) be a pair such that $\mathcal{X} = \{X_1, X_2, \dots, X_\ell\}$ is a collection of subsets of vertices of a graph G , called *bags*, and T is a tree whose nodes are in one-to-one correspondence with the elements of \mathcal{X} . When this creates no ambiguity, X_i will denote both a bag of \mathcal{X} and the node of T whose corresponding bag is X_i . The pair (\mathcal{X}, T) is a *tree-decomposition* of G if: (i) for every edge (u, v) of G , there exists a bag X_i that contains both u and v , and (ii) for every vertex v of G , the set of nodes of T whose bags contain v induces a non-empty (connected) subtree of T . The *width* of (\mathcal{X}, T) is $\max_{i=1}^{\ell} \{|X_i| - 1\}$, while the *treewidth* of G is the minimum width over all tree-decompositions of G . For an n -vertex graph of treewidth t , a tree-decomposition of width t can be found in FPT time [32].

Definition 1 A tree-decomposition (\mathcal{X}, T) of a graph G is called *nice* if T is a rooted tree with the following properties [33].

- (P.1) Any node of T has at most two children.
- (P.2) If a node X_i of T has two children whose bags are X_j and $X_{j'}$, then $X_i = X_j = X_{j'}$. In this case, X_i is a *join bag*.
- (P.3) If a node X_i of T has only one child X_j , then $X_i \neq X_j$ and there exists a vertex $v \in G$ such that either $X_i = X_j \cup \{v\}$ or $X_i \cup \{v\} = X_j$. In the former case X_i is an *introduce bag*, while in the latter case X_i is a *forget bag*.
- (P.4) If a node X_i is a leaf of T , then X_i contains exactly one vertex, and X_i is a *leaf bag*.

Note that, given a tree-decomposition of width t , a nice tree-decomposition can be computed in $O(t \cdot n)$ time (see, e.g., [34]).

Fig. 2 **a** Inessential intersection vertices, and **b** a twin-pair



3 Basic Properties of Map Graphs and Their Witnesses

The following statements have already been discussed in the work by Chen et al. [13], even though in a weaker or different form. For completeness, we provide full proofs.

Let $G = (V, E)$ be a map graph and let $W = (V \cup I, A)$ be a witness of G , i.e., W is a planar bipartite graph such that $W^2[V] = G$. A vertex $u \in I$ is an *intersection vertex* of W , while a vertex $v \in V$ is a *real vertex* of W . Also, we let $n_V = |V|$, $n_I = |I|$, and $n = n_V + n_I$.

Property 1 A graph is a k -map graph if and only if it admits a witness such that the maximum degree of every intersection vertex is k .

Property 2 A graph G admits a map if and only if each of its biconnected components admits a map. Also, if G admits a hole-free map, then G is biconnected.

Let $W = (V \cup I, A)$ be an embedded witness (i.e., with a prescribed combinatorial embedding). An intersection vertex $u \in I$ is *inessential* if $\deg(u) = 2$ and there exists $u' \in I$ such that $N(u) \subset N(u')$; see Fig 2a. Furthermore, a pair of intersection vertices $u_1, u_2 \in I$ is a *twin-pair* if $N(u_1) = N(u_2) = \{v, w\}$, for some $v, w \in V$, and W contains a face whose boundary consists of a single closed walk with exactly four edges with end-vertices v, u_1, w, u_2 ; see Fig 2b. Note that removing an inessential vertex or one vertex of a twin-pair from W does not modify $W^2[V]$.

Definition 2 An embedded witness of a map graph is *compact* if it contains neither inessential intersection vertices nor twin-pairs.

We remark that a compact witness is not necessarily minimal, i.e., it may contain intersection vertices of degree greater than two whose removal does not modify its half-square; see also [13]. However, in our setting, removing further information from a witness would have an impact on the proof of Theorem 2 and on the recognition algorithm (Sect. 5).

The next lemma shows that focusing on compact witnesses is not restrictive.

Lemma 1 *tharcompact* A graph $G = (V, E)$ is a map graph if and only if it admits a compact witness. Also, G is a k -map graph if and only if it admits a compact witness whose intersection vertices have degree at most k .

Proof For the first part of the statement, recall that a graph admits a map if and only if it admits a witness [2, 13]. Thus, if G has a compact witness, it is a map graph. For

the other direction, suppose that G admits a map. Let \hat{W} be any embedded witness of G . Let W be the embedded graph obtained from \hat{W} by removing all inessential intersection vertices and by iteratively removing one intersection vertex for each twin-pair. Since we only removed degree-2 intersection vertices whose neighbors are already incident to a common intersection vertex, it holds that $W^2[V] = \hat{W}^2[V]$. Thus, since $\hat{W}^2[V] = G$, it holds $W^2[V] = G$.

For the second part of the statement, recall that G admits a k -map if and only if it has a witness whose intersection vertices have degree at most k , by Property 1. Moreover, we have seen before that, for every witness \hat{W} , there exists (at least) one compactwitness W obtained by fixing a combinatorial embedding of \hat{W} and by possibly removing intersection vertices of degree 2 that are either inessential or part of a twin-pair. Since \hat{W} is bipartite, this implies that any intersection vertex that belongs to both \hat{W} and W has the same degree in the two graphs. \square

Given a graph G such that $n_V \geq 3$ and a map \mathcal{M} of G , the *order* of a point $p \in \mathcal{M}$, denoted by $ord(p)$, is equal to the number of nations and holes whose boundary contains p . Let $W = (V \cup I, A)$ be the bipartite embedded graph computed with the *compactconstruction*, defined as follows. For ease of description, we define W by constructing a topological embedding of it; refer to Fig 1. In particular, the witness of Fig 1c is compactand constructed with the described procedure (which again follows the lines of the work in [13]). For each nation $\mathcal{M}(v)$, we place the real vertex v in its interior. For each point p such that $ord(p) \geq 3$, we add an intersection vertex u_p to I and place it at point p . We connect each real vertex v to the intersection vertices that lie on the boundary of $\mathcal{M}(v)$, by drawing crossing-free simple arcs inside $\mathcal{M}(v)$. Note that, for each intersection vertex u_p , it holds $deg(u_p) = ord(p) - h(p)$, where $h(p)$ is the number of holes in \mathcal{M} whose boundary contains p . Finally, we remove inessential intersection vertices and, iteratively, a vertex for each twin-pair. For instance, in Fig 1b the nations colored light-yellow and light-green share two order-3 points that would give rise to inessential intersection vertices, which are indeed not reported in Fig 1c.

Lemma 2 *Let $W = (V \cup I, A)$ be the embedded graph obtained from the map \mathcal{M} of G by means of the compactconstruction. Then W is a compactwitness of G .*

Proof The fact that the compactconstruction defines a topological embedding of W , and in particular that each arc is simple and no two arcs intersect at an interior point, follows by construction. Moreover, we explicitly removed inessential intersection vertices and twin-pairs, if any. So, it remains to prove that $W^2[V] = G$. By construction, for each edge of $W^2[V]$, there is an edge in G . Also, any edge of G is represented by at least one point of \mathcal{M} whose order is at least two. Since we created an intersection vertex for each point of order greater than two, it remains to argue about points of order exactly two. Any such a point is an interior point of a simple arc along which two nations $\mathcal{M}(v)$ and $\mathcal{M}(w)$ touch. The two endpoints of this arc must have order at least three, which implies that edge (v, w) exists in $W^2[V]$. \square

In [13], it is observed (without a formal argument) that a map graph is hole-free if and only if it admits a witness whose faces have 4 or 6 edges each. The next characterization improves over this observation and hence can be of independent interest. A connected

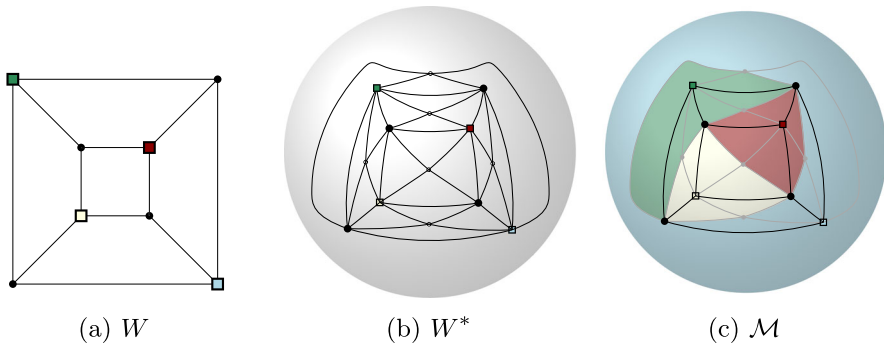


Fig. 3 Illustration for the proof of Theorem 2 (Color figure online)

embedded graph is a *quadrangulation* if each face boundary consists of a single closed walk with 4 edges.

Theorem 2 *A graph is a hole-free map graph if and only if it admits a compactwitness that is a biconnected quadrangulation.*

Proof (\Leftarrow) Refer to Fig 3. If a graph G admits a compactwitness W , then by Theorem 1 G is a map graph. Thus we only need to show that W yields a map that is hole-free, by exploiting the assumption that W is a biconnected quadrangulation. Let W^* be the embedded graph defined as follows: We add a dummy vertex inside each face of W and connect it to all vertices on the boundary of the face. Since W is biconnected, each face boundary is a simple cycle, and therefore W^* is an embedded triangulation, i.e., each face contains three edges on its boundary. Let Γ be a topological embedding on the sphere of W^* . It follows that each triangular face of Γ is incident to exactly one real vertex of W . Also, for each real vertex v of W , the union of the triangular faces incident to v defines a region R_v that contains v in its interior. The latter property of Γ allows us to construct a map \mathcal{M} of G by setting $\mathcal{M}(v)$ to be equal to the closure of R_v , for each real vertex v of W . The fact that \mathcal{M} is a hole-free map follows by construction. Namely, all points of the sphere are covered by nations, hence there are no holes. Also, the points of \mathcal{M} of order at least three are in a one-to-one correspondence with the intersection vertices of W , and any other order-2 point of \mathcal{M} lies along a simple arc connecting two points of higher order.

(\Rightarrow) Let \mathcal{M} be a hole-free map of a graph G . By Property 2, G is biconnected. Let W be the compactwitness of G computed by the compactconstruction from \mathcal{M} ; for instance, the compactwitness in Fig 3a is constructed from the map in Fig 3c by using the compactconstruction. Since G is biconnected, W is (at least) connected, and thus the boundary of any face of W consists of a single closed walk. In the following we prove that W is a quadrangulation. This, together with the fact that W is simple, implies biconnectivity. Since W is connected, simple and bipartite, the boundary of each of its faces consists of a single closed walk containing at least four edges. Assume for a contradiction that W contains a face f with more than four edges on the closed walk π defining its boundary. Then, since W is bipartite, π contains at least six edges. Consider any intersection vertex u on π . Let v and w be the two (real) vertices that

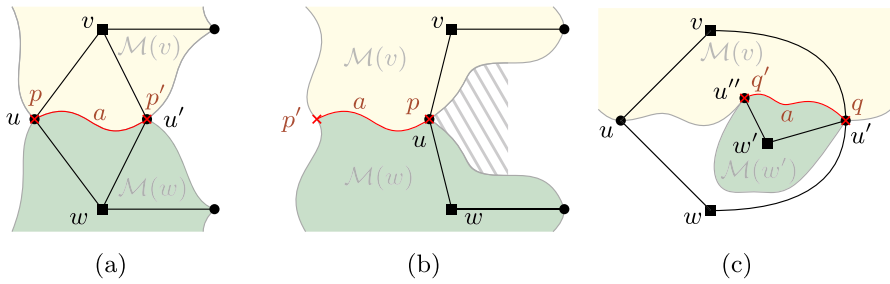


Fig. 4 Illustrations for the proof of Theorem 2

precede and follow u along π , respectively. We distinguish two cases based on whether u is or is not the only intersection vertex in π that is adjacent to both v and w .

Suppose first that u is the only intersection vertex in π that is adjacent to both v and w . By construction, u has been placed on a point $p \in \mathcal{M}$ such that $ord(p) \geq 3$. Since \mathcal{M} is hole-free, p is the endpoint of a simple arc a that forms a shared boundary between $\mathcal{M}(v)$ and $\mathcal{M}(w)$. Let p' be the other endpoint of a . Since $ord(p') \geq 3$ and \mathcal{M} is hole-free, again W contains an intersection vertex u' that has been placed on p' . Either u' belongs to π or not. In the first case, we contradict the fact that u is the only intersection vertex of π adjacent to both v and w ; see Fig. 4a for an illustration. In the second case, p is on the boundary of a hole, which contradicts the fact that \mathcal{M} is hole-free; see Fig. 4b for an illustration.

Suppose now that there is another intersection vertex u' in π adjacent to both v and w ; refer to Fig. 4c for an illustration. Since u and u' are both adjacent to v and w and both belong to π , in order for π to contain six (or more) edges, at least one of v, u, w, u' occurs more than once along π , and between its (at least) two occurrences, there must be a real vertex w' . Note that, by definition of v and w , such vertex occurring multiple times along π cannot be u . We assume that such vertex is u' , the remaining cases can be handled with symmetric arguments. Consider a traversal of π that visits v, u, w in this order. Up to a renaming of the vertices, we can assume that w' is the vertex that precedes the last occurrence of u' in this traversal. Let q be the point of \mathcal{M} where u' has been placed on. By construction, q is the endpoint of a simple arc b that forms a shared boundary between $\mathcal{M}(w')$ and $\mathcal{M}(v)$. Let q' be the other endpoint of b . Since $ord(q') \geq 3$ and \mathcal{M} is hole-free, again W contains an intersection vertex u'' that has been placed on q' and that belongs to π by construction. Since u'' is adjacent to both v and w' , we get a contradiction to the fact that the last occurrence of u' before v is encountered after w' in the above traversal. \square

Lemma 3 *A (hole-free) map graph G admits a compactwitness with $n \leq 6n_V - 10$ (respectively, $n \leq 3n_V - 4$) vertices.*

Proof Suppose first that G is hole-free. Let $W = (V \cup I, A)$ be a compactwitness of G that is a quadrangulation, which exists by Theorem 2. We start with the following claim.

Claim 1 $\forall u \in I$, it holds $deg(u) > 2$.

Proof Suppose, for a contradiction, that W contains an intersection vertex u such that $\deg(u) = 2$ and let f be any face of W that contains u on its boundary. Let u' be the other intersection vertex on the boundary of f , and observe that $N(u) \subseteq N(u')$. If $N(u) = N(u')$, then u and u' form a twin-pair. Otherwise, u is inessential. Both cases contradict the fact that W is compact. \square

Since W is crossing-free and bipartite (because W is a witness), it holds $|E| = \sum_{u \in I} \deg(u) \leq 2(n_V + n_I) - 4$. By Claim 1, we have $\sum_{u \in I} \deg(u) \geq 3n_I$. Putting all together, we have $2(n_V + n_I) - 4 \geq \sum_{u \in I} \deg(u) \geq 3n_I$. Consequently, $n_I \leq 2n_V - 4$ and thus $n = n_V + n_I \leq 3n_V - 4$.

Suppose now that G is not hole-free. Let W be any compactwitness of G and let W' be the graph obtained by removing all degree-2 intersection vertices from W . Since W' is crossing-free and bipartite and since its intersection vertices have degree at least 3, we can conclude as above that W' has at most $3n_V - 4$ vertices. We now claim that W has no more than $3n_V - 6$ degree-2 intersection vertices, which concludes the proof, since these are exactly the vertices in $W \setminus W'$. To prove the claim, replace each degree-2 intersection vertex of W with an edge connecting its two neighbors. Since W is crossing-free, the resulting graph W^* is also crossing-free. Also, each intersection vertex in W^* has degree greater than two. Since W contains no twin-pairs, W^* contains no pairs of homotopic parallel edges. Thus, Euler's formula for planar graphs still applies and therefore W^* contains at most $3n_V - 6$ edges. Since each edge of W^* corresponds to at most one degree-2 intersection vertex of W , the claim follows. \square

Based on Lemma 3, we can make the following remark.

Remark 1 Without loss of generality, we assume in the following that any compactwitness W of G has $n \leq 3n_V - 4$ vertices if G is hole-free, or $n \leq 6n_V - 10$ vertices otherwise.

4 Embedding Sketches

Let G be an input graph. Property 2 allows us to assume that G is biconnected, and thus every witness of G , if any, is connected. Also, by Theorem 1, it suffices to consider compactwitnesses.

Let (\mathcal{X}, T) be a nice tree-decomposition of G of width $t = \omega - 1$, i.e., each bag contains at most ω vertices. Given a bag $X \in \mathcal{X}$, we denote by T_X the subtree of T rooted at X , and by $G_X = (V_X, E_X)$ the subgraph of G induced by all the vertices in all bags of T_X . Let $W_X = (V_X \cup I_X, A_X)$ be a compactwitness of G_X (in particular, $W_X^2[V_X] = G_X$). Note that, although G is connected, G_X may have multiple connected components. However, since G is connected, each connected component of G_X must contain at least one vertex of X . Moreover, for each connected component C of G_X , there is a connected component C' of W_X such that C' is a witness of C . A vertex of W_X is an *anchor vertex* if it is either a real vertex of X or an intersection vertex whose neighbors in W_X all belong to X . Observe that if an intersection vertex u has a neighbor v in $V_X \setminus X$, then no real vertex in $V \setminus V_X$ is adjacent to v , and therefore

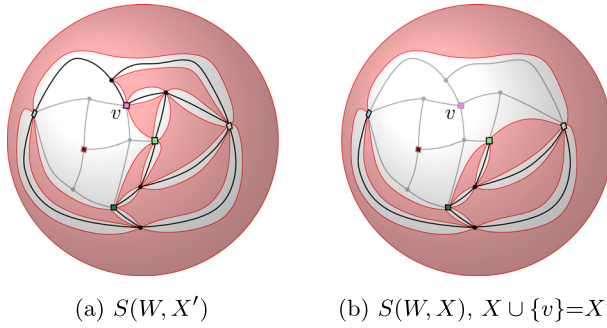
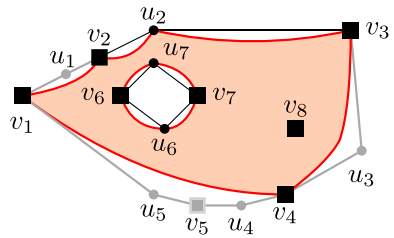


Fig. 5 **a** A sketch $S(W, X')$ computed from the witness W of Fig 1 with respect to a bag X' ($V_{X'} = V$). The anchor vertices of X' are opaque, while the non-anchor vertices are faded. The active boundaries are red and the background of the active faces is light red. **b** A sketch $S(W, X)$, where $X \cup \{v\} = X'$ computed from $S(W, X')$ by applying the deletion operation (Sect. 5) (Color figure online)

Fig. 6 An active boundary (red) made of three closed walks (edges are omitted):
 $\langle v_1, v_2, u_2, v_3, v_4, v_1 \rangle$,
 $\langle u_6, v_6, u_7, v_7 \rangle$, $\langle v_8 \rangle$; vertices u_1, u_3, u_4, u_5, u_5 have been shortcut (Color figure online)



there is no way to add further edges to u without creating a false adjacency involving v .

We will exploit anchor vertices to reduce the size of W_X from $O(|V_X|)$ to $O(\omega)$, by “sketching” parts of the embedding that are not relevant.¹ The idea of sketching an embedded graph is inspired by a previous work about orthogonal planarity [29]; applying this idea to our problem requires the development of several new tools and concepts, described in the remainder of this section (and partly in Sect. 2). A face f of W_X is *active* either if its boundary contains only one vertex v (which implies $W_X = (\{v\}, \emptyset)$) and v is an anchor vertex, or if its boundary contains more vertices among which there are at least two anchor vertices; refer to Fig 5a. The *active boundary* of f (red in Fig 5a) is obtained by shortcutting all non-anchor vertices of f , where the *shortcut operation* is defined as follows. For a closed walk π and a vertex v in π , shortcutting v consists of removing each occurrence of v (if more than one), together with the edge (u, v) that precedes it in π , and the edge (v, u') that follows it in π , and of adding the edge (u, u') between u and u' in π . Figure 6 illustrates a single face f and the corresponding active boundary. The *embedding sketch* (for short the sketch) of W_X with respect to X is the embedded graph $S(W_X, X)$ formed by all the vertices and edges that belong to the active boundaries of W_X . For each active boundary B_f of an active face f of W_X , $S(W_X, X)$ has an *active face* f^* (light red in Figs. 5a and

¹ In the database and data engineering fields, sketching algorithms form a powerful toolkit to compress data in a way that supports answering various queries [35]. Our idea of sketching has some similarities with this concept but serves a different purpose.

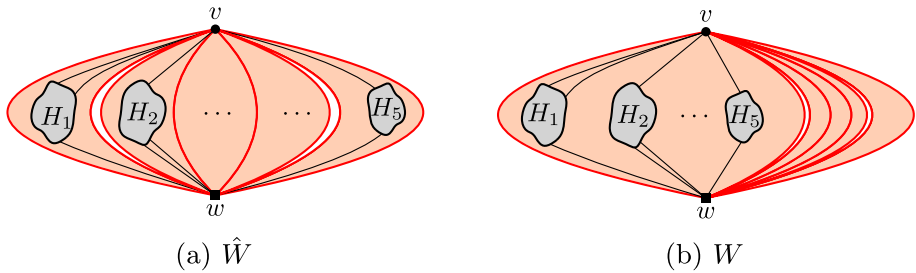


Fig. 7 Illustrations for the proof of Lemma 4. Modifying the rotation system of \hat{W} such that each H_i lies in B_1 and all other non-extensible active boundaries become empty

6). Note that $S(W_X, X)$ also has faces that are not active (white in Figs. 5a and 6). Also, the position system of W_X yields a position system for $S(W_X, X)$, since if two closed walks of distinct components of W_X were incident to the same active face f , then the two corresponding closed walks of $S(W_X, X)$ are also incident to the same active face f^* . However, $S(W_X, X)$ may not be bipartite any longer (as in Fig. 6) and it may contain multiple edges (but no self-loops). It is worth noting that the embedding sketchof W_X can be defined with respect to any bag X' , as long as $V_{X'} = V_X$ (see Fig. 5a).

We now further refine $S(W_X, X)$ to avoid active boundaries that are not useful for our purposes. Namely, an active boundary is *non-extensible* if it consists of two homotopic parallel edges. Given a witness W of G , the *restriction* of W to G_X is the compactwitness $W[G_X]$ of G_X obtained from W by removing all the real vertices not in G_X , all the intersection vertices that are isolated (due to the removal of some real vertices) or inessential, as well as a vertex for each twin-pair until the graph contains none of them. The next lemmas allow us to bound the size of a sketch.

Lemma 4 *If G is a map graph, then it admits a compactwitness W with the following property. If $S(W[G_X], X)$ contains $h > 1$ non-extensible active boundaries that share the same pair of end-vertices v and w , then the vertices of W lie in at most one of these h active boundaries.*

Proof Refer to Fig. 7. Let \hat{W} be a compactwitness of G . Suppose $S(\hat{W}[G_X], X)$ contains $h > 1$ non-extensible active boundaries, denoted by B_1, B_2, \dots, B_h . Also, denote by v and w the common end-vertices of these boundaries. Let H_i be the subgraph of \hat{W} that lies inside B_i (if any), for $1 \leq i \leq h$. Since each B_i consists of two parallel edges, v and w separate H_i and $S(W[G_X], X) \setminus H_i$. We obtain a new compactwitness W of G by modifying the rotation system of \hat{W} so that each H_i lies inside B_1 . \square

Remark 2 By Lemma 4, we assume in the following that for any compactwitness W of G such that, for some $X \in \mathcal{X}$, the sketch $S(W[G_X], X)$ contains $h > 1$ non-extensible active boundaries, the vertices of W lie in at most one of such active boundaries. Therefore, in $S(W[G_X], X)$, we keep only one of the corresponding h pairs of homotopic parallel edges.

Lemma 5 *A sketch $S(W_X, X)$ contains $O(\omega)$ vertices and edges.*

Proof With a similar argument as in the proof of Lemma 3 we can show that, in W_X , each real vertex in X is adjacent to $O(\omega)$ intersection vertices that are anchor vertices. Therefore, $S(W_X, X)$ contains $O(\omega)$ vertices in total. Concerning the number of edges, since $S(W_X, X)$ is embedded on the sphere, it contains $O(\omega)$ edges such that each pair of edges is either non-parallel or non-homotopic parallel. In addition, since each of these edges participates in at most one homotopic pair by Remark 2, it follows that $S(W_X, X)$ contains $O(\omega)$ edges. \square

We now exploit the concept of a sketch to define an equivalence relation among witnesses.

Definition 3 Two compact witnesses W_X and W'_X of G_X are X -equivalent if they have the same sketch with respect to X , i.e., $S(W_X, X) = S(W'_X, X)$.

The next lemma deals with the size of the quotient of such a relation.

Lemma 6 *The X -equivalence relation yields $\omega^{O(\omega)}$ classes for the compact witnesses of G_X .*

Proof Let n_1 be the number of possible (abstract) graphs that can be obtained from the real vertices of X and all possible sets of intersection vertices. For any such graph, let n_2 be the maximum number of its possible rotation and position systems. It follows that the number of X -equivalent classes is upper bounded by the product of n_1 and n_2 .

Given the set X of real vertices and a compact witness W_X of G_X , any sketch $S(W_X, X)$ contains $O(\omega)$ intersection vertices, as otherwise W_X would contain inessential intersection vertices or twin-pairs. Since each intersection vertex is adjacent to a set of at most ω real vertices, we can bound the number n_{int} of possible sets of intersection vertices by $a \cdot \sum_{i=2}^{\omega} \binom{\omega}{i} < a \cdot 2^{\omega}$, where a is the maximum number of intersection vertices in any sketch that have the same set of neighbors. Since $a \in O(\omega)$, we have that $n_{\text{int}} \in 2^{O(\omega)}$. Let I_X be one of the n_{int} possible sets of intersection vertices. The number n_{abs} of distinct abstract graphs with vertex set $X \cup I_X$ can be upper bounded by the number of possible neighborhoods of a real vertex combined for all real vertices, that is

$$n_{\text{abs}} \leq \prod_{v \in X} \omega^{\deg(v)} = \omega^{\sum_{v \in X} \deg(v)} \leq \omega^{O(\omega)}$$

holds, which yields $n_1 \leq n_{\text{int}} \cdot n_{\text{abs}} \in \omega^{O(\omega)}$.

For a fixed graph S , the number of possible rotation systems n_{rot} is upper bounded by the number of possible permutations of edges around each vertex. Thus we have

$$n_{\text{rot}} \leq \prod_{v \in S} \deg(v)! < \prod_{v \in S} \deg(v)^{\deg(v)} \leq \omega^{O(\sum_{v \in S} \deg(v))} \leq \omega^{O(\omega)}.$$

Each rotation system of S fixes the closed walk of each face of each connected component of S . Since S contains, over all its connected components, at most ω closed walks (at most one for each real vertex in X) and hence at most ω faces, for the number n_{pos} of possible position systems it holds $n_{\text{pos}} \leq \omega^{\omega}$. Therefore we have $n_2 \leq n_{\text{rot}} \cdot n_{\text{pos}} \in \omega^{O(\omega)}$, which yields $n_1 \cdot n_2 \in \omega^{O(\omega)}$, as desired. \square

5 Algorithmic Framework

Let $G = (V, E)$ be an input graph, let k be an integer, and let (\mathcal{X}, T) be a nice tree-decomposition of G of width $t = \omega - 1$. We present an algorithmic framework to test whether G is a k -map graph or a hole-free k -map graph. Namely, we traverse T bottom-up and equip each bag $X \in \mathcal{X}$ with a suitably defined set of sketches, called *record* and denoted by R_X . The framework can be tailored by imposing different properties for the records. The next three properties are rather general; the first two are useful to prove the correctness of our approach, as shown in Theorem 3, whereas the third comes into play when dealing with the efficiency of the approach, and in particular in Lemma 7.

Definition 4 The record R_X is *feasible* if the following properties hold:

- F1 For every compactwitness W_X of G_X , R_X contains its sketch $S(W_X, X)$.
- F2 For every entry $r \in R_X$, there is a compactwitness W_X of G_X such that $r = S(W_X, X)$.
- F3 R_X contains no duplicates.

Lemma 7 Any $X \in \mathcal{X}$ contains $\omega^{O(\omega)}$ entries, each of size $O(\omega)$, if R_X is feasible.

Proof By FF1–FF3, the entries of R_X are all and only the possible sketches of W_X and are all distinct. Hence, $|R_X| \in \omega^{O(\omega)}$ by Lemma 6. Each sketch has size $O(\omega)$ by Lemma 5. \square

We now describe the additional properties that we incorporate in the framework. In order to verify that G admits a k -map we exploit Property 1, which translates into verifying that, for each sketch, the degree of any intersection vertex is at most k .

Definition 5 A record R_X is *k -map feasible* if it is feasible and it contains a non-empty subset $R_X^* \subseteq R_X$, called *subrecord*, for which the following additional property holds:

- F4 For every entry $r \in R_X$, it holds $r \in R_X^*$ if and only if r contains no intersection vertex u with $\deg(u) > k$.

It is worth observing that, since an intersection vertex of degree k implies the existence of a clique of size k in the input graph G , property FF4 is trivially verified when $k \geq \omega$. On the other hand, the size of the largest clique of a k -map graph is $\lfloor 3k/2 \rfloor$ (see, e.g., [13]).

To check whether G has a hole-free k -map, we exploit Theorem 2. Namely, consider a sketch $S(W_X, X)$ and an active boundary B_f of $S(W_X, X)$. Let f be the active face of W_X corresponding to B_f . Note that any edge e that is part of B_f represents a subsequence π_e of a closed walk π in the boundary of f . Therefore, to control the number of edges on the boundary of each face of W_X , for every edge e that is part of an active boundary of $S(W_X, X)$ we also store a counter $c(e) \geq 1$, which represents the number of edges in π_e . If there is an edge e such that $c(e) > 4$, then G does not admit a compactwitness W that is a quadrangulation such that $W_X = W[G_X]$; hence we can avoid storing counters greater than four. Moreover, for any face f of a compact witness W of G , we know that there exist two bags \hat{X}' and \hat{X} in T such that \hat{X}' is the child

of \hat{X} , \hat{X} is a forget bag, the active boundary representing f in \hat{X}' has more than one anchor vertex, while the one in \hat{X} has only one anchor vertex (and hence is not part of $S(W_{\hat{X}}, \hat{X})$). We call such an active boundary *complete* in \hat{X}' , as it will not be modified anymore by the algorithm. As such, for each complete active boundary, the sum of the counters of its edges in $S(W_{\hat{X}'}, \hat{X}')$ must be exactly 4, otherwise G does not admit a compactwitness W that is a biconnected quadrangulation such that $W_{\hat{X}} = W[G_{\hat{X}}]$.

Definition 6 A record R_X is *hole-free feasible* if it is feasible and it contains a non-empty subset $R_X^\circ \subseteq R_X$, called *subrecord*, for which the following additional property holds:

- F5 For every entry $r \in R_X$, it holds $r \in R_X^\circ$ if and only if r contains no intersection vertex u with $\deg(u) > k$ and each complete active boundary of r (if any) is such that its edge counters sum up to 4.

Each leaf bag contains only one vertex v , thus its record consists of one sketch with only one active face whose active boundary is $\langle v \rangle$. Such a record can be computed in $O(1)$ time and it is trivially feasible. Also, it is hole-free (and hence k -map) feasible, as its unique active boundary is not complete. The next three operations are performed on a non-leaf bag X of T , based on the type of X , to compute a k -map or hole-free feasible record R_X , if any.

Deletion operation. Let X be a forget bag whose child X' in T has a (hole-free) k -map feasible record $R_{X'}$. Let v be the vertex forgotten by X . We generate R_X from $R_{X'}$ as follows.

For a fixed sketch $S(W_{X'}, X')$ of $R_{X'}$, let $N_I(v) \subseteq N(v)$ be the set of intersection vertices adjacent to v in $S(W_{X'}, X')$. Since v is forgotten by X , all its neighbors have already been processed, thus no vertex in $N_I(v)$ can connect vertices that will be introduced by bags visited after X . Therefore, for every vertex $y \in N_I(v) \cup \{v\}$ and for every sketch $S(W_{X'}, X')$ of $R_{X'}$, we apply a *deletion operation*, which consists of updating every active boundary B_f of $S(W_{X'}, X')$ containing y ; see Fig. 5b. Namely, let B_f be one of these active boundaries, we distinguish two cases based on whether B_f contains only y or it contains further vertices. Let π_y be the closed walk of B_f that contains all occurrences of y (there might be more than one). If B_f contains only y , we remove π_y (and hence the whole active boundary B_f) from $S(W_{X'}, X')$. If B_f contains further vertices, we shortcut every occurrence of y in π_y . Also for every edge e introduced to shortcut y such that e replaces edges e_1 and e_2 of π_y , we set $c(e) = c(e_1) + c(e_2)$. Observe that, if y has only one neighbor u in π_y , this procedure creates a self-loop at u , which we remove. If this procedure generates more than one pair of homotopic parallel edges with the same pair of end-vertices, then we keep only one such pair. Once all active boundaries have been updated, the resulting embedded graph is stored in R_X . After every sketch of $R_{X'}$ has been processed, we might have produced the same embedded graph for R_X from two distinct sketches of $R_{X'}$; in this case we keep only one copy.

Addition operation. Let X be an introduce bag whose child X' in T has a (hole-free) k -map feasible record $R_{X'}$. Let v be the vertex introduced by X and $N_X(v) \subseteq N(v)$ be the set of vertices that are neighbors of v and belong to X . We generate R_X from

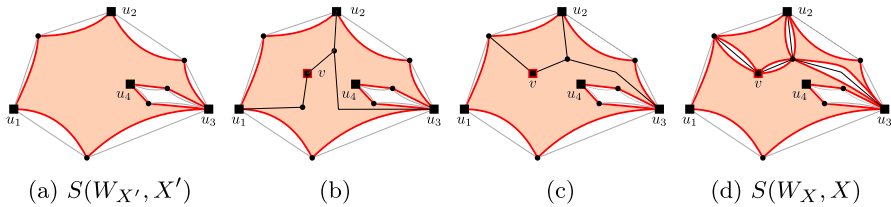


Fig. 8 Illustration for the addition of vertex v . **a** Details of a face of $S(W_{X'}, X')$ that contains all the neighbors of v . **b–c** Two distinct embedded graphs computed from $S(W_{X'}, X')$ by introducing vertex v in different ways. **d** The sketch $S(W_X, X)$ obtained by replacing the active boundary of the red face with the new active boundaries corresponding to the three newly created active faces in **c** (Color figure online)

$R_{X'}$ with the following *addition operation*. For each sketch $S(W_{X'}, X')$ of $R_{X'}$, the high-level idea is to exhaustively generate all possible embedded graphs that can be obtained by introducing v in $S(W_{X'}, X')$. We distinguish two cases.

Case 1: $N_X(v) = \emptyset$. For each active boundary B_f of $S(W_{X'}, X')$, we generate a new embedded graph by adding the closed walk $\langle v \rangle$ to B_f .

Case 2: $N_X(v) \neq \emptyset$. We look for a face f^* of $S(W_{X'}, X')$ that contains all the vertices of $N_X(v)$ on its active boundary B_f (which may consist of multiple closed walks). If such a face does not exist, we discard $S(W_{X'}, X')$. Else, for any such face, we generate a set of entries E_{f^*} as follows. Intuitively, we will insert v inside f^* and generate one entry of E_{f^*} for each possible way in which v can be connected to its neighbors. Namely, we can connect v to its neighbors by means of different intersection vertices and by realizing different permutations of the edges around v and around those neighbors that appear multiple times along some closed walk of B_f ; refer to Fig. 8 for an illustration. Concerning the intersection vertices, we can use those that already belong to B_f and are adjacent only to vertices in $N_X(v)$, as well as we can create new ones. We note that since v has at most $\omega - 1$ neighbors in $N_X(v)$, there are $\sum_{i=1}^{\omega-1} \binom{\omega-1}{i} = 2^{\omega-1}$ possible combinations of intersection vertices (see also the proof of Lemma 6). This is done avoiding inessential intersection vertices and twin-pairs. For each choice of intersection vertices, since the degree of a vertex is $O(\omega)$, there are $\omega^{O(\omega)}$ distinct rotation systems to consider. Additionally, if B_f consists of multiple closed walks, we shall consider all possible permutations of the edges around v that do not cause edge crossings (i.e., any edge permutation in which there are no four edges e_1, e_2, e_3, e_4 in this order around v , such that e_1, e_3 connect v to the vertices of a closed walk π and e_2, e_4 connect v to the vertices of a closed walk π' with $\pi \neq \pi'$), and we consider each of them independently as a new embedded graph. Based on the fixed intersection vertices and rotation system, if the insertion of v does not split f^* into multiple faces, we can suitably update B_f , otherwise we can generate the new active boundaries that appear in place of B_f ; see in particular Fig. 8d. Also, for each newly introduced edge e in a closed walk, we set $c(e) = 1$.

Merge operation. Let X be a join bag whose children X_1 and X_2 in T have (hole-free) k -map feasible records R_{X_1} and R_{X_2} , respectively. We generate R_X from R_{X_1} and R_{X_2} . Since X is a join bag, X, X_1 , and X_2 contain the same vertices, whereas G_{X_1} and G_{X_2} only share the vertices in X . Consider any pair of sketches $S(W_{X_1}, X)$ of R_{X_1} and

$S(W_{X_2}, X)$ of R_{X_2} . Such sketches share the same set of real vertices, whereas they may have different sets of intersection vertices and different combinatorial embeddings. At high-level, we aim at combining $S(W_{X_1}, X)$ and $S(W_{X_2}, X)$ in all possible ways, provided that the original rotation and position systems of each sketch are preserved and that we never insert a subgraph of one sketch into a non-active face of the other. In practice, we apply the *merge operation*, consisting of the next steps.

- (S.1) We compute all possible unions of the two abstract graphs underlying the two sketches. Namely, let I_{X_1} and I_{X_2} be the sets of intersection vertices of $S(W_{X_1}, X)$ and $S(W_{X_2}, X)$, respectively. We identify each pair of real vertices the two sketches share, and we consider all possible abstract graphs whose set of intersection vertices I_X is such that: (a) $I_X \subseteq I_{X_1} \cup I_{X_2}$; (b) for each intersection vertex of I_{X_1} there is an intersection vertex in I_X with the same set of neighbors, and the same holds for I_{X_2} .
- (S.2) For each generated graph S^* , we compute all combinatorial embeddings, i.e., all possible rotation and position systems yielding a topological embedding on the sphere of S^* . If no such combinatorial embedding exists, we discard S^* , else we go to the next step.
- (S.3) We generate all possible one-to-one mappings ϕ_1 between intersection vertices of S^* and of $S(W_{X_1}, X)$, and all possible one-to-one mappings ϕ_2 between intersection vertices of S^* and of $S(W_{X_2}, X)$.
- (S.4) We check, for each pair ϕ_1, ϕ_2 , that the restriction of the resulting embedded graph on the real vertices, intersection vertices (up to the mapping defined by ϕ_1 and ϕ_2) and edges of each of the two sketches preserves the corresponding rotation and position systems. If so, we go to the next step; otherwise, we discard the candidate solution.
- (S.5) Since the previous step guaranteed that the active boundaries of each sketch are preserved when looking at the corresponding restriction, we can verify that there is no subgraph of one sketch inside a non-active face of the other.
- (S.6) We suitably update the active boundaries of the resulting embedded graph and we add it to R_X . More precisely, the boundary of a face is active if it does not correspond to a non-active boundary in any of the two sketches and it contains either exactly one anchor vertex or at least two anchor vertices.
- (S.7) We remove inessential intersection vertices and iteratively one intersection vertex for each twin-pair, until there are no twin-pairs.
- (S.8) Once all pairs of sketches have been processed, we remove possible duplicates.

This concludes the description of the main algorithmic steps for proving Theorem 1. Next, we provide lemmas to establish the correctness and the time complexity of these steps.

Lemma 8 *Let X be a forget bag whose child X' in T has a k -map (resp. hole-free) feasible record $R_{X'}$. The algorithm either rejects the instance or computes a k -map (resp. hole-free) feasible record R_X of X in $\omega^{O(\omega)}$ time.*

Proof Let v be the vertex forgotten by X . We prove that the record R_X generated by applying the deletion operation is feasible, given that $R_{X'}$ is feasible. In particular, since we removed possible duplicates, FF3 holds and it remains to argue about FF1 and FF2.

To this aim, since X is a forget bag, note that $G_X = G_{X'}$. Hence any compactwitness $W_{X'}$ of $G_{X'}$ is also a compactwitness of G_X . Moreover, since $R_{X'}$ is feasible, it follows by **FF1** that $R_{X'}$ contains a sketch $S(W_{X'}, X')$ for every compactwitness $W_{X'}$. Now, since $X' = X \cup \{v\}$, the sketch of $W_{X'}$ with respect to X , namely $S(W_{X'}, X)$, coincides with the one obtained by applying the deletion operation to $S(W_{X'}, X')$. Thus, **FF1** holds for X . Similarly, since $R_{X'}$ is feasible, it follows by **FF2** that every entry of $R_{X'}$ is the sketch $S(W_{X'}, X')$ of a compactwitness $W_{X'}$ of $G_{X'}$. Again since $X' = X \cup \{v\}$, the entry of R_X obtained by applying the deletion operation to $S(W_{X'}, X')$ corresponds to the sketch $S(W_{X'}, X)$. Thus, **FF2** holds for X and consequently R_X is feasible, as claimed.

Suppose now that $R_{X'}$ is k -map feasible, i.e., $R_{X'}^* \neq \emptyset$. We show how to check whether a sketch of R_X belongs to R_X^* . Since the deletion operation does not modify the degree of any intersection vertex, the subrecord R_X^* contains all sketches of R_X generated from sketches in $R_{X'}^*$. Based on this observation, we can check whether $R_X^* = \emptyset$ or not. In the former case the algorithm rejects the instance, in the latter case R_X is k -map feasible. Suppose that $R_{X'}$ is hole-free feasible, i.e., $R_{X'}^\circ \neq \emptyset$. Again the subrecord R_X° contains all sketches of R_X that have been generated from sketches in $R_{X'}^\circ$, and that contain no active boundary whose edge counters sum up to 4. To decide whether an active boundary is complete, it suffices to check whether the parent of X is a forget bag such that the shortcuttings due to the removal of the forgotten vertex make that active boundary a self-loop. If any complete active boundary does not meet this condition, the corresponding sketch does not belong to R_X° . As before if $R_X^\circ = \emptyset$ the algorithm rejects the instance, otherwise R_X is hole-free feasible.

By Lemma 7, $R_{X'}$ contains $\omega^{O(\omega)}$ entries, each of size $O(\omega)$. Updating each of them takes $O(\omega)$ time. Also, R_X contains at most as many entries as $R_{X'}$. It follows that removing duplicates can be naively done in $(\omega^{O(\omega)})^2 \in \omega^{O(\omega)}$ time. For the sake of efficiency, if we interpret each rotation and position system together as a number with $\tilde{O}(\omega^2)$ bits, then removing duplicates can be done in $\tilde{O}(\omega^2) \cdot \omega^{O(\omega)} \in \omega^{O(\omega)}$ time by using radix sort (we omit the details as the asymptotic running time would be the same). We have seen that condition **FF4** is always verified. Checking condition **FF5** requires scanning each active boundary in R_X and decide whether it is complete or not, and if so to verify whether it will become a self-loop when visiting the parent of X . This can be done in $O(\omega)$ time for each of the $O(\omega)$ active boundaries of each of the $\omega^{O(\omega)}$ sketches, and thus in $\omega^{O(\omega)}$ time overall. Thus R_X and its subrecords can be computed in $\omega^{O(\omega)}$ time, as desired. \square

Lemma 9 *Let X be an introduce bag whose child X' in T has a k -map (resp. hole-free) feasible record $R_{X'}$. The algorithm either rejects the instance or computes a k -map (resp. hole-free) feasible record R_X of X in $\omega^{O(\omega)}$ time.*

Proof Let v be the vertex introduced by X . We prove that the record R_X generated by applying the addition operation is feasible, given that $R_{X'}$ is feasible. Regarding **FF1**, let $W_{X'}$ and W_X be a witness of $G_{X'}$ and G_X , respectively, such that $W_X[G_{X'}] = W_{X'}$. Since **FF1** holds for $R_{X'}$, we know that $S(W_{X'}, X') \in R_{X'}$. Observe that the only difference between W_X and $W_{X'}$ lies in the presence of vertex v and of a (possibly empty) set I_v of intersection vertices adjacent to v .

If $N_X(v) = \emptyset$, then v forms a trivial closed walk that might be added in any face of $W_{X'}$ that either consists of exactly one anchor vertex or contains at least two anchor vertices (among possibly other non-anchor vertices). We recall that an active face satisfying the mentioned properties corresponds to an active boundary of the witness' sketch. Also, adding the closed walk to a face that contains more than one vertex, but at most one anchor vertex, on its boundary would imply that the resulting witness cannot be augmented to a witness of G , since G is biconnected. Since Case 1 places v in all possible active boundaries of $S(W_{X'}, X')$, we can conclude that $S(W_X, X)$ belongs to R_X .

On the other hand, if $N_X(v) \neq \emptyset$, then all neighbors of v belong to a common boundary of some face f of $W_{X'}$, as otherwise the rotation system of W_X would not be compatible with a topological embedding (in particular, some edges would cross each other). Hence all neighbors of v are part of the same active boundary B_f of $S(W_{X'}, X')$. Since Case 2 exhaustively considers all the ways in which v can be inserted into B_f , avoiding inessential intersection vertices and twin-pairs (which cannot belong to W_X , since it is compact), we can again conclude that $S(W_X, X)$ belongs to R_X . Consequently, FF1 holds for R_X .

About FF2, it suffices to prove that each entry generated by the addition operation is indeed a sketch of some compact witness of G_X with respect to X . Since FF2 holds for $R_{X'}$, the addition operation starts from a sketch $S(W_{X'}, X')$ and generates new entries in which there are neither inessential intersection vertices nor twin-pairs; therefore, such entries are indeed sketches of compact witnesses, as desired.

Concerning FF3, if R_X contained two entries r_1, r_2 that are the same (up to a homeomorphism of the sphere), then r_1 and r_2 would have been originated by the same sketch r of $R_{X'}$, as otherwise either r_1 and r_2 would not be the same or FF3 would not hold for $R_{X'}$. On the other hand, since the addition operation inserts v in different ways but without repetitions, it cannot generate two entries that are the same starting from a single entry of $R_{X'}$. Thus, FF3 holds for R_X .

If $R_{X'}$ is k -map feasible, we know that R_X^* contains those sketches of $R_{X'}$ for which the addition operation did not introduce intersection vertices of degree larger than k . Based on this observation, we can check whether $R_X^* = \emptyset$ or not. In the former case the algorithm rejects the instance, in the latter case R_X is k -map feasible. The case when $R_{X'}$ is hole-free feasible can be proved analogously as in the proof of Lemma 8.

Finally, each single entry constructed by the addition operation can be computed in $O(\omega)$ time and R_X contains $\omega^{O(\omega)}$ entries by Lemma 7. Also, condition FF4 can be easily verified in $O(\omega)$ time, for each of the $\omega^{O(\omega)}$ sketches of R_X . Checking condition FF5 requires scanning each active boundary in R_X and deciding whether it is complete or not. This can be done in $O(\omega)$ time, for each of the $O(\omega)$ active boundaries of each of the $\omega^{O(\omega)}$ sketches, and thus in overall $\omega^{O(\omega)}$ time. Thus, R_X and its subrecords can be computed in $\omega^{O(\omega)}$ time. □

The proof of the next lemma exploits the merge operation.

Lemma 10 *Let X be a join bag whose children X_1 and X_2 in T both have k -map (resp. hole-free) feasible records R_{X_1} and R_{X_2} . The algorithm either rejects the instance or computes a k -map (resp. hole-free) feasible record R_X of X in $\omega^{O(\omega)}$ time.*

Proof We prove that the record R_X generated by applying the merge operation is feasible, given that R_{X_1} and R_{X_2} are feasible. Consider any compactwitness W_X of G_X and its restrictions $W_X[G_{X_1}]$ and $W_X[G_{X_2}]$ to G_{X_1} and G_{X_2} , respectively. By definition of restriction, there must exist a mapping of the intersection vertices of W_X to the intersection vertices of $W_X[G_{X_1}]$ such that when looking at the restriction of W_X to the real and intersection vertices of $W_X[G_{X_1}]$ (up to the above mentioned mapping), the rotation and position systems of $W_X[G_{X_1}]$ are preserved. The same property must hold for $W_X[G_{X_2}]$. These properties clearly carry over to the corresponding sketches $S(W_X, X)$, $S(W_X[G_{X_1}], X)$, and $S(W_X[G_{X_2}], X)$. Since R_{X_1} and R_{X_2} are feasible, they contain $S(W_X[G_{X_1}], X)$ and $S(W_X[G_{X_2}], X)$, respectively. Hence, Steps S.(S.1)–S.(S.4) guarantee that the aforementioned mapping is considered and that all the above properties hold on the candidate solutions given by the combination of $S(W_X[G_{X_1}], X)$ and $S(W_X[G_{X_2}], X)$. Moreover, any subgraph of W_X that belongs to $W_X[G_{X_1}]$ but not to $W_X[G_{X_2}]$, except for the shared vertices of X , must lie in an active face of $W_X[G_{X_2}]$ (and vice versa); if this is not the case, then W_X would not be augmentable to a witness of G , since G is biconnected. This property translates into verifying that any subgraph of $S(W_X[G_{X_1}], X)$ lies in an active face of $S(W_X[G_{X_2}], X)$ (and vice versa). This is achieved in Step S.(S.5). Step S.(S.6) suitably updates the active boundaries so that a boundary is active only if it represents a face of W_X that either consists of exactly one anchor vertex or contains at least two anchor vertices, as by definition of active boundary. Step S.(S.7) removes inessential intersection vertices and twin-pairs, which is a safe operation because W_X is compact. Therefore, we can conclude that $S(W_X, X)$ belongs to R_X , and thus FF1 holds for R_X . Concerning FF2, any entry S in R_X generated by the merge operation, starting from entries $S(W_{X_1}, X) \in R_{X_1}$ and $S(W_{X_2}, X) \in R_{X_2}$, defines a way to combine the combinatorial embeddings of $S(W_{X_1}, X)$ and $S(W_{X_2}, X)$ at common real vertices and at possibly common (based on some mappings ϕ_1 and ϕ_2) intersection vertices. Such information can be used to combine in the same way the corresponding witnesses W_{X_1} and W_{X_2} , which exist because FF2 holds for R_{X_1} and R_{X_2} , respectively. On the other hand, such combination yields a compactwitness W_X of G_X with respect to X , whose sketch is S , as desired. Thus FF2 holds for R_X . In Step S.(S.8) we remove possible duplicates, hence FF3 holds by construction for R_X . Therefore R_X is feasible. Since the merge operation does not increase the degree of intersection vertices, and since R_{X_1} and R_{X_2} are k -map feasible, the subrecord R_X^* contains all sketches of R_X generated from sketches in $R_{X_1}^*$ and $R_{X_2}^*$. If $R_X^* = \emptyset$, the algorithm rejects the instance, otherwise R_X is k -map feasible. If R_{X_1} and R_{X_2} are hole-free feasible, R_X° contains all sketches of R_X that are generated from sketches in $R_{X_1}^\circ$ and $R_{X_2}^\circ$, and whose complete active boundaries are such that the edge counters sum up to 4. If $R_X^\circ = \emptyset$, the algorithm rejects the instance, otherwise R_X is hole-free feasible.

Concerning the time complexity, we process each pair of sketches, one in R_{X_1} and one in R_{X_2} , and since both R_{X_1} and R_{X_2} are feasible, we have $\omega^{O(\omega)}$ such pairs. Each of Steps S.(S.1), S.(S.2), and S.(S.3) generates $\omega^{O(\omega)}$ new entries, and each entry is computed in $O(\omega)$ time. The remaining steps all run in $O(\omega)$ time for each processed entry. Condition FF4 can be easily verified in $O(\omega)$ time, for each of the $\omega^{O(\omega)}$ sketches of R_X . Furthermore, verifying condition FF5 requires scanning the active boundaries of each entry in R_X and deciding whether it is complete or not. This

can also be done in $O(\omega)$ time for each of the $O(\omega)$ active boundaries of each of the $\omega^{O(\omega)}$ sketches, and thus in overall $\omega^{O(\omega)}$ time. Consequently, R_X and its subrecords can be computed in $\omega^{O(\omega)}$ time. \square

Lemmas 8–10 imply the next theorem, which summarizes the correctness of the approach.

Theorem 3 *Let G be a graph in input to the algorithm, along with a nice tree-decomposition (T, \mathcal{X}) of G and an integer $k > 0$. Graph G is a k -map graph, respectively a hole-free k -map graph, if and only if the algorithm reaches the root ρ of T and the record R_ρ is k -map feasible, respectively hole-free feasible.*

We are finally ready to prove Theorem 1. We recall that if $k \geq n - 1$, recognizing n -vertex (resp. hole-free) k -map graphs coincides with recognizing general n -vertex (resp. hole-free) map graphs.

Proof of Theorem 1 We first discuss the decision version of the problem for a fixed $k > 0$. Namely, the algorithm described below is used in a binary search to find the optimal value of k . Recall that t is the width of the tree decomposition (i.e., $\omega = t + 1$). Note that, if G is a positive instance, then k varies in the range $[1, t + 1]$, since the size of the largest clique of G is at most $t + 1$. Thus the algorithm is executed $O(\log t)$ times, which however does not affect the asymptotic running time.

If G is not biconnected, by Property 2, it is not a hole-free map graph, and it is a k -map graph if and only if all its biconnected components are k -map graphs. Hence we run our algorithm on each biconnected component independently. Theorem 3 implies the correctness of the algorithm (which assumes the input graph to be biconnected).

For the time complexity, suppose that G has $h \geq 1$ biconnected components and let n_i be the size of the i -th component C_i , for each $i \leq h$. Decomposing G into its biconnected components takes $O(n + m)$ time [36], where m is the number of edges of G and, since G has treewidth t , it holds $m \in O(n \cdot t^2)$. Given a tree-decomposition of G with $O(n)$ nodes and width t , we can easily derive a tree-decomposition (T_i, \mathcal{X}_i) for each C_i in overall $O(n)$ time, such that each T_i has $O(n_i)$ nodes and width at most t . Then we can apply the algorithm in [33] to obtain, in $O(n_i)$ -time, a nice tree-decomposition of C_i with $O(n_i)$ nodes without increasing the original width. Since each bag is processed in $t^{O(t)}$ time by Lemmas 8–10, the algorithm runs in $t^{O(t)} \cdot n_i$ time for each C_i . Since $\sum_{i=1}^h n_i \in O(n)$, decomposing the graph and applying the algorithm to all its biconnected components takes $t^{O(t)} \cdot n$ time.

To reconstruct a witness of a yes-instance, we store additional pointers for each record (a common practice in dynamic programming). Namely, for each sketch S of a record R_X of a bag X , we store a pointer to the sketch of the child bag X' that generated S , if X is an introduce or forget bag, and we store two pointers to the two sketches of the children bags X_1 and X_2 that generated S , if X is a join bag. With these pointers at hand, we can apply a top-down traversal of T , starting at any sketch of the non-empty subrecord of ρ , and reconstruct the corresponding witness W by incrementally combining the retrieved sketches, except at forget bags (the only points in which we lose information). Suppose first that G is a k -map graph but not hole-free. If G is not biconnected, a witness W^* of G is obtained by merging the

witnesses of its biconnected components. Note that distinct witnesses corresponding to distinct biconnected components of G can only share real vertices. Thus, each intersection vertex of W^* has degree at most k and W^* is a certificate by Property 1. Suppose now that G is a hole-free k -map graph. Then G is biconnected and the resulting witness is a biconnected quadrangulation whose intersection vertices have degree at most k , a certificate by Theorem 2. \square

6 Conclusions and Open Problems

We have shown how to recognize (hole-free) k -map graphs in linear time for input graphs having bounded treewidth. The general problem of recognizing map graphs efficiently remains a major algorithmic challenge. To restrict the complexity of the input, further parameters of interest might be the cluster vertex deletion number [37] and the clique-width [38] of the input graph, as well as the treewidth of the putative witness [16].

Another interesting line of research would be generalizing our framework to recognize (g, k) -map graphs, i.e., those graphs that admit a k -map on a surface of genus g (see, e.g., [3]).

We finally recall that the complexity of recognizing (hole-free) k -map graphs is open for any fixed $k \geq 5$. A natural step in this direction is hence studying the complexity of recognizing (hole-free) 5-map graphs.

Acknowledgements We acknowledge the anonymous reviewers of this paper for their valuable suggestions.

Author Contributions All authors equally contributed to the research described in this manuscript and to its preparation.

Funding Open access funding provided by Università degli Studi di Perugia within the CRUI-CARE Agreement. Fabrizio Montecchiani acknowledges funding by MUR of Italy, under PRIN Project n. 2022ME9Z78 - NextGRAAL: Next-generation algorithms for constrained GRAPh visuALization.

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Monadic Second-Order Logic Formulation

We prove that the problem of recognizing map graphs can be expressed by using MSO_2 logic, which implies the existence of a fixed-parameter tractable algorithm for parameterized by treewidth.

Theorem 4 *Given an n -vertex graph G of treewidth t , there is an algorithm that decides whether G is a map graph in time $f(t) \cdot O(n)$, for some computable function f .*

Proof Let V and E be the vertex and edge set of G , respectively. We construct a graph $G^* = (V \cup C, E \cup D)$ by augmenting G . For every subset S of V such that S forms a clique in G : (i) We add a vertex $v_S \in C$ to G^* , and (ii) We add an edge $(v_S, u) \in D$ for each $u \in S$. Since G has n vertices and treewidth t , it admits a tree-decomposition T with $O(n)$ bags, such that each bag contains at most t vertices. Also, for any clique of G there is a bag that contains all its vertices. Altogether, it follows that G contains $O(2^t n)$ cliques and hence G^* has $O(2^t n)$ vertices. Moreover, the treewidth of G^* is at most $t + 1$. Namely, we can obtain a valid tree-decomposition T^* of G^* from T as follows. For a vertex $v_S \in C$, let S be the corresponding clique in V and let ν be a bag of T that contains all the vertices of S . For any such a vertex v_S , we add a new leaf bag ν^* in T^* , connected only to ν and containing v_S and all the vertices in S . It is immediate to verify that T^* is a tree-decomposition of G^* of width at most $t + 1$.

By construction, G is a map graph if and only if there exists a subset B of D such that: (i) The graph G_B formed by the edges of B is planar (note that it is bipartite by construction), and (ii) For every edge (u, v) of E , there is a path between u and v in G^* composed of two edges of B . Indeed, if B exists, then G_B is a witness of G . Both conditions (planarity and the existence of a length-2 path) can be expressed in MSO_2 logic.² Consequently, the statement follows by Courcelle's theorem [27]. \square

We remark that the proof of Theorem 4 can be easily modified to find the minimum k such that G is a k -map graph. Namely, for the decision version of the problem, it suffices to add a vertex $v_S \in C$ to G^* only if the clique S has size at most k . However, it is less obvious how to adjust the proof in order to test whether G is also hole-free, in particular, how to additionally ensure that G_B has a planar embedding in which all faces have length at most six [13].

References

1. Angelini, P., Bekos, M.A., Da Lozzo, G., Gronemann, M., Montecchiani, F., Tappini, A.: Recognizing map graphs of bounded treewidth. In: SWAT. LIPIcs, vol. 227, pp. 8–1818. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Tórshavn, Faroe Islands (2022)
2. Chen, Z., Grigni, M., Papadimitriou, C.H.: Planar map graphs. In: STOC, pp. 514–523. ACM, Dallas, Texas, USA (1998)
3. Dujmovic, V., Eppstein, D., Wood, D.R.: Structure of graphs with locally restricted crossings. SIAM J. Discret. Math. **31**(2), 805–824 (2017)

² B. Courcelle and J. Engelfriet. Graph Structure and Monadic Second-Order Logic—A Language-Theoretic Approach, volume 138 of Encyclopedia of mathematics and its applications. Cambridge University Press, 2012.

4. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Intersection-link representations of graphs. *J. Graph Algorithms Appl.* **21**(4), 731–755 (2017)
5. Brandenburg, F.J.: Characterizing 5-map graphs by 2-fan-crossing graphs. *Discret. Appl. Math.* **268**, 10–20 (2019)
6. Brandenburg, F.J.: Characterizing and recognizing 4-map graphs. *Algorithmica* **81**(5), 1818–1843 (2019)
7. Chen, Z., He, X., Kao, M.: Nonplanar topological inference and political-map graphs. In: SODA, pp. 195–204. ACM/SIAM, Baltimore, Maryland, USA (1999)
8. Chen, Z.: Approximation algorithms for independent sets in map graphs. *J. Algorithms* **41**(1), 20–40 (2001)
9. Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Fixed-parameter algorithms for (k, r) -center in planar graphs and map graphs. *ACM Trans. Algorithms* **1**(1), 33–47 (2005)
10. Fomin, F.V., Lokshtanov, D., Misra, N., Saurabh, S.: Planar f -deletion: Approximation, kernelization and optimal FPT algorithms. In: FOCS, pp. 470–479. IEEE, New Brunswick, NJ, USA (2012)
11. Fomin, F.V., Lokshtanov, D., Panolan, F., Saurabh, S., Zehavi, M.: Decomposition of map graphs with applications. In: ICALP. LIPIcs, vol. 132, pp. 60–16015. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
12. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Bidimensionality and geometric graphs. In: SODA, pp. 1563–1575. SIAM, Kyoto, Japan (2012)
13. Chen, Z., Grigni, M., Papadimitriou, C.H.: Map graphs. *J. ACM* **49**(2), 127–138 (2002).
14. Thorup, M.: Map graphs in polynomial time. In: FOCS, pp. 396–405. IEEE, Palo Alto, California, USA (1998)
15. Chen, Z., Grigni, M., Papadimitriou, C.H.: Recognizing hole-free 4-map graphs in cubic time. *Algorithmica* **45**(2), 227–262 (2006)
16. Mnich, M., Rutter, I., Schmidt, J.M.: Linear-time recognition of map graphs with outerplanar witness. *Discret. Optim.* **28**, 63–77 (2018)
17. Chen, Z.: New bounds on the edge number of a k -map graph. *J. Graph Theory* **55**(4), 267–290 (2007)
18. Dujmović, V., Joret, G., Micek, P., Morin, P., Ueckerdt, T., Wood, D.R.: Planar graphs have bounded queue-number. *J. ACM* **67**(4), 22–12238 (2020)
19. Bekos, M.A., Da Lozzo, G., Hliněný, P., Kaufmann, M.: Graph product structure for h -framed graphs. In: ISAAC. LIPIcs, vol. 248, pp. 23–12315. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Seoul, Korea (2022)
20. Brandenburg, F.J.: Book embeddings of k -map graphs. *CoRR* **abs/2012.06874** (2020)
21. Le, H., Le, V.B.: Map graphs having witnesses of large girth. *Theor. Comput. Sci.* **772**, 143–148 (2019)
22. Le, H., Le, V.B.: Constrained representations of map graphs and half-squares. In: MFCS. LIPIcs, vol. 138, pp. 13–11315. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Aachen, Germany (2019)
23. Le, H., Le, V.B.: Hardness and structural results for half-squares of restricted tree convex bipartite graphs. *Algorithmica* **81**(11–12), 4258–4274 (2019)
24. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. *SIAM J. Comput.* **6**(3), 505–517 (1977)
25. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York, USA (1999)
26. Robertson, N., Seymour, P.D.: Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms* **7**(3), 309–322 (1986)
27. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inf. Comput.* **85**(1), 12–75 (1990)
28. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015)
29. Di Giacomo, E., Liotta, G., Montecchiani, F.: Orthogonal planarity testing of bounded treewidth graphs. *J. Comput. Syst. Sci.* **125**, 129–148 (2022)
30. Jansen, B.M.P., Lokshtanov, D., Saurabh, S.: A near-optimal planarization algorithm. In: SODA, pp. 1802–1811. SIAM, Oregon, USA (2014)
31. Kociumaka, T., Pilipczuk, M.: Deleting vertices to graphs of bounded genus. *Algorithmica* **81**(9), 3655–3691 (2019)
32. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25**(6), 1305–1317 (1996)

33. Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms* **21**(2), 358–402 (1996)
34. Kloks, T.: *Treewidth, Computations and Approximations*. LNCS, vol. 842. Springer, Germany (1994)
35. Cormode, G.: Data sketching. *ACM Queue* **15**(2), 60 (2017)
36. Tarjan, R.E., Vishkin, U.: Finding biconnected components and computing tree functions in logarithmic parallel time (extended summary). In: FOCS, pp. 12–20. IEEE, West Palm Beach, Florida, USA (1984)
37. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. *Theory Comput. Syst.* **47**(1), 196–217 (2010)
38. Courcelle, B., Engelfriet, J., Rozenberg, G.: Handle-rewriting hypergraph grammars. *J. Comput. Syst. Sci.* **46**(2), 218–270 (1993)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

**Patrizio Angelini¹ · Michael A. Bekos² · Giordano Da Lozzo³ ·
Martin Gronemann⁴ · Fabrizio Montecchiani⁵ · Alessandra Tappini⁵**

Patrizio Angelini
pangelini@johncabot.edu

Michael A. Bekos
bekos@uoi.gr

Giordano Da Lozzo
giordano.dalozzo@uniroma3.it

Martin Gronemann
mgronemann@ac.tuwien.ac.at

Alessandra Tappini
alessandra.tappini@unipg.it

¹ Department of Mathematics, Natural, and Applied Sciences, John Cabot University, Rome, Italy

² Department of Mathematics, University of Ioannina, Ioannina, Greece

³ Department of Engineering, Roma Tre University, Rome, Italy

⁴ Algorithms and Complexity Group, TU Wien, Vienna, Austria

⁵ Department of Engineering, University of Perugia, Perugia, Italy