Modification to Planarity is Fixed Parameter Tractable

Fedor V. Fomin, Petr A. Golovach, Dimitrios M. Thilikos

Modification to Planarity is Fixed Parameter Tractable

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway
fomin@ii.uib.no

Petr A. Golovach
Department of Informatics, University of Bergen, Norway
petr.golovach@uib.no

Dimitrios M. Thilikos
AlGCo project-team, LIRMM, Université de Montpellier, CNRS, Montpellier, France
sedthilk@thilikos.info

Abstract

A *replacement action* is a function $\mathcal{L}$ that maps each k-vertex labeled graph to another k-vertex graph. We consider a general family of graph modification problems, called $\mathcal{L}$-Replacement to $\mathcal{C}$, where the input is a graph G and the question is whether it is possible to replace in G some k-vertex subgraph H of it by $\mathcal{L}(H)$ so that the new graph belongs to the graph class $\mathcal{C}$. $\mathcal{L}$-Replacement to $\mathcal{C}$ can simulate several modification operations such as edge addition, edge removal, edge editing, and diverse completion and superposition operations. In this paper, we prove that for any action $\mathcal{L}$, if $\mathcal{C}$ is the class of planar graphs, there is an algorithm that solves $\mathcal{L}$-Replacement to $\mathcal{C}$ in $O(|G|^2)$ steps. We also present several applications of our approach to related problems.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Modification problems, Planar graphs, Irrelevant vertex technique

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.28

Funding *Fedor V. Fomin*: Supported by the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL". Supported by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.
Petr A. Golovach: Supported by the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL". Supported by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.
Dimitrios M. Thilikos: Supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010). Supported by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.

1 Introduction

The *irrelevant vertex technique* was proposed by Robertson and Seymour for the Disjoint Paths problem [24, 25], which is the central algorithmic result of their Graph Minors series of papers. A superficial description of the technique is

> *If the treewidth of the input graph is small, then standard techniques on graphs of bounded treewidth can be used to solve the problem we have on hands. Otherwise, the graph contains an irrelevant vertex, that is, the vertex whose removal does not change the problem.*

Of course, the devil is in details, and usually in order to make the irrelevant vertex technique work, highly non-trivial arguments are involved in proving the existence of an irrelevant vertex, see e.g. [1, 7, 12, 13, 14, 15, 17, 18, 19, 20, 22, 23]. We also refer to [6, Chapter 7] for a high-level overview of the irrelevant vertex technique.

36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019).
Editors: Rolf Niedermeier and Christophe Paul; Article No. 28; pp. 28:1–28:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are a number of generic algorithmic results in the literature explaining why and when a certain algorithmic technique is successful. For example, we know that many problems can be solved by dynamic programming on graphs of bounded treewidth, and Courcelle's theorem explains why this happens [5]. However, we do not know any generic characterization of problems solvable by the irrelevant vertex technique and the quest for such a characterization is the main motivation for this paper.

We show that the irrelevant vertex approach can be used to establish fixed-parameter tractability of a very general class of graph transformation problems. The problem we consider is the following: For some integer k, is it possible to transform an input graph G into a planar graph by performing at most k allowed changes? The allowed changes are defined through the set of the following *replacement actions*. Suppose that for every labelled k-vertex graph H we have a list $L(H)$ of labelled k-vertex graphs. Then the replacement action selects a subset of k vertices X in the graph G and replaces the subgraph $G[X]$ induced by X by a graph F from the list $L(G[X]))$. More precisely, the action selects a k-sized vertex subset X of G labelled by numbers $\{1,\ldots,k\}$ and, given that H is the labelled k-vertex graph obtained from $G[X]$, we select a labelled k-vertex graph F from $L(H)$ and replace H by F. Thus the vertex set of the new graph G' is $V(G)$ and it has the same adjacencies as in G except pairs of vertices from X. In the transformed graph, vertices $u, v \in X$ labelled by $i, j \in \{1,\ldots,k\}$ are adjacent in G' if and only if $\{i, j\}$ is an edge of F. Then the task is for a given graph G and the family of allowed replacement actions, to decide whether there is a replacement action transforming G into a planar graph.

The problem of replacements to a planar graph encompasses many interesting graph modification problems. For example, the simplest replacement action is defined by associating with every k-vertex graph H the list $L(H)$ consisting of an edgeless k-vertex graph. This encodes the problem of finding in graph G a set of k vertices X such that deleting all edges with both endpoints in X results in a planar graph. By selecting an appropriate set of replacements, one can encode many interesting graph transformation problems, with specified properties of the replaced subgraph. This also includes various structural properties of replaced subgraph H, like being a a matching, a clique, or a cycle. Similarly replacement actions can describe the structural properties of the replacement graph F. For example, the condition could be that F is the complement of H. Or it could be some quantitive property, like if we delete $k/100$ edges, we have to add a least $k/200$ edges, or that one graph is obtained from another by flipping k edges according to some specified rules, etc.

Our main result is an algorithm that for any choice of the replacement actions on k-vertex graphs, decides whether an n-vertex input graph G can be made planar by making use of replacement in time $\mathcal{O}(f(k) \cdot n^2)$, where f is some function of k only. In other words, the problem is fixed-parameter tractable (FPT) parameterized by k.

While, from the general perspective, the proof of our main result follows the path of all irrelevant vertex techniques papers, there are several significant differences compared with the previous works. The main difficulty we have to resolve is the following. The most common argument towards application of the irrelevant vertex technique is that if we find a large "flat wall" (a grid-like part of the graph which has a planar embedding), then the central part of the wall is irrelevant. This does not work for replacements – the reason is that replacements are *non-local* and may affect vertices that are anywhere in the graph. Thus even if a vertex is inside a huge wall, it still can be used in the action. To overcome this issue we have to define an equivalence relation between pieces of the wall expressing the fact that equivalent pieces are "interchangeable" with respect to any application of an action. Next we have to detect a sufficiently large set of "equivalent" pieces of the wall and prove that at least one of these pieces can be considered untouched by the transformation. Identifying equivalent parts of the wall is the main technical challenge. While parts of the wall are of bounded treewidth, typical applications of equivalence relations for graphs of bounded treewidth used in the literature strongly exploit the property that the boundary of treewidth bounded graphs is also bounded. However, this is not true in our case because the parts of the wall may have huge boundaries. This requires a new way to define equivalence relations and an efficient procedure for handling new relations, which in turn allows us to simplify the input of large treewidth. We believe that the technical contribution of our work will be the starting point for dealing with other algorithmic applications on planar graphs.

Section 3 is devoted to a high-level description of our algorithm, while the formal definitions for some basic concepts of this description are presented in Section 4. In Section 5 we provide several examples of problems that can be reduced to the replacement framework and in Section 6 we conclude with some open problems and further research directions.

Related work. Graph planarization and, more generally, graph modification, is one of the main themes in Parameterized Algorithms. For example, Planar Vertex Deletion, where one asked to remove at most k vertices of the input graph to make it planar. This problem is fixed-parameter tractable parameterized by k by the generic result from the Graph Minors project of Robertson and Seymour [24], who showed that every minor-closed property of graphs can be checked in polynomial time. The result of Robertson and Seymour yields the existence of an algorithm for this problem but provides no way to construct such an algorithm. Constructive algorithms for Planar Vertex Deletion were considered in [16, 23]. The fastest known algorithm for this problem runs in time $2^{\mathcal{O}(k\log k)}n$ [15]. The problem of obtaining a planar graph by contracting edges was considered in [13]. More generally, the problem of modifying a graph to some graph class excluding some fixed minors, was considered in [9, 8, 21, 10, 11].

2 Definition of the problem and outline of the algorithm

Elementary definitions. We use $\mathbb{N}$ to denote the set of all non-negative numbers. Given a $k_1, k_2 \in \mathbb{N}$, we denote by $[k_1, k_2]$ the set $\{k_1, \dots, k_2\}$ and given k we denote $[k] = [1, k]$. Given a function $\varphi : A \to B$ and a subset $X \subseteq A$ we naturally extend φ by using $\varphi(X)$ to denote $\{\varphi(x) \mid x \in X\}$. We also write $\varphi|_X$ to denote the restriction of φ to $X \subseteq A$. We denote by $\mathbf{inj}(A, B)$ the set of all injections from A to B. For $\varphi \in \mathbf{inj}(A, B)$, we denote by φ^{-1} the mapping of $\varphi(B)$ to A that is the reverse of φ.

All graphs in this paper are undirected, finite, simple, and without multiple edges. Given a graph G we denote by $V(G)$ and $V(E)$ the set of its vertices and edges, respectively. We also denote $|G| = |V(G)|$. If $S \subseteq V(G)$, then we denote by $G \setminus S$ the graph obtained by G after removing from it all vertices in S, together with the incident edges. We define the subgraph of G *induced* by S as the graph $G[S] = G \setminus (V(G) \setminus S)$. Given an induced subgraph G' of G, we define $G \setminus G' = (V(G), E(G) \setminus E(G'))$, i.e., $G \setminus G'$ is obtained from G if we remove the edges of G'. Given two graphs G_1 and G_2, we define their *union* as $G_1 \cup G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$.

2.1 Replacement actions

Replacements. A *k-numbered-graph* is any graph H where $V(H) = [k]$, i.e., the vertices of H are the numbers $\{1, \dots, k\}$. We denote the set of all k-numbered graphs by $\mathcal{H}_k$ and we set $\mathcal{H} = \bigcup_{k\in\mathbb{N}} \mathcal{H}_k$. A *replacement action* (abbreviated as *R-action*) is any function $\mathcal{L} : \mathcal{H} \to \mathcal{H}$, where for every $H \in \mathcal{H}$, $|\mathcal{L}(H)| = |H|$, i.e., graphs in $\mathcal{H}$ are mapped to same-size graphs.

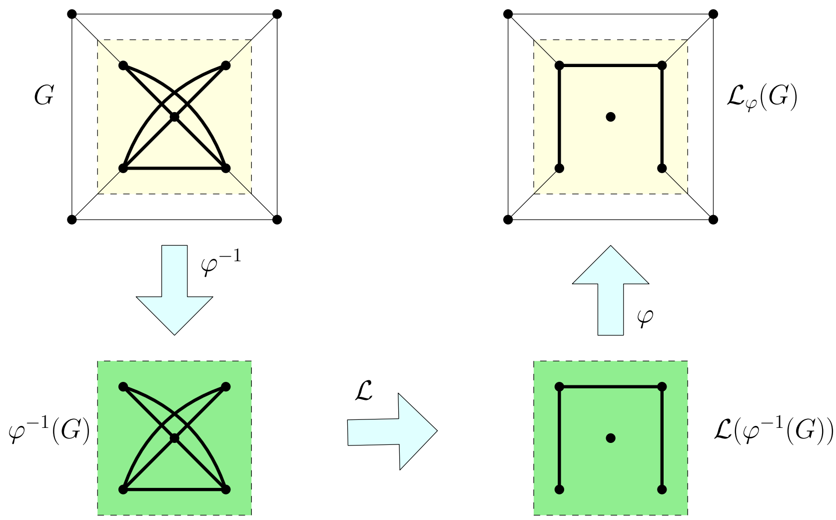

Figure 1 An illustration of a replacement action where the R-action $\mathcal{L}$ replaces graphs by their complements.

Let G be a graph and let $\varphi \in \mathbf{inj}([k], V(G))$. We define $\varphi^{-1}(G) = ([k], \{\varphi^{-1}(e) \mid e \in E(G[\varphi([k])])\})$, i.e., we see $\varphi^{-1}(G)$ is the graph in $\mathcal{H}_k$ that is isomorphic, via φ, to the subgraph of G where φ applies.

Let G be a graph and let G' be a graph where $V(G') \subseteq V(G)$. We denote $G \sqcup G' = (G \setminus G[V(G')]) \cup G'$, i.e., $G \sqcup G'$ occurs if we remove from G the edges between vertices in G' and then add all edges of G'. Given a graph G, a $\varphi \in \mathbf{inj}([k], V(G))$, and a $H \in \mathcal{H}_k$, we define $\varphi(H) = \{\varphi([k]), \{\varphi(e) \mid e \in E(H)\}\}$. Given an R-action $\mathcal{L} : \mathcal{H} \to \mathcal{H}$, we set $\mathcal{L}_\varphi(G) = G \sqcup \varphi(\mathcal{L}(\varphi^{-1}(G)))$, in other words, we consider the part of G that is delimited by φ and then we replace this part by its image via $\mathcal{L}$ (see Figure 1 for an example).

We now have all ingredients we need for defining our general problem.

Replacement to planarity. We examine the following family of problems, that may vary, depending on the choice of the R-action $\mathcal{L}$:

> $\mathcal{L}$-REPLACEMENT TO A PLANAR GRAPH. ($\mathcal{L}$-RP)
> **Input:** A graph G and a non-negative integer k.
> **Question:** Is there a $\varphi \in \mathbf{inj}([k], V(G))$ such that $\mathcal{L}_\varphi(G)$ is planar?

▶ **Theorem 1.** *For every R-action $\mathcal{L}$, there exists an algorithm that given an instance (G, k) of $\mathcal{L}$-REPLACEMENT TO A PLANAR GRAPH, reports whether (G, k) is a yes-instance in $O_k(|G|^2)$ steps*[1].

The main result of the paper is a proof of Theorem 1. In fact, we give an algorithm that runs in the same running time, for the following more general annotated version of this problem:

[1] Given a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and a function $g : \mathbb{N} \to \mathbb{N}$, we use the notation $f(k, n) = O_k(g(n))$ in order to say that there is a function $h : \mathbb{N} \to \mathbb{N}$ where $f(k, n) = O(h(k) \cdot g(n))$.

$\mathcal{L}$-ANNOTATED REPLACEMENT TO A PLANAR GRAPH ($\mathcal{L}$-ARP)
Input: A graph G, a set of *annotated* vertices $R \subseteq V(G)$, and a non-negative integer k.
Question: Is there a $\varphi \in \mathbf{inj}([k], R)$ such that $\mathcal{L}_\varphi(G)$ is planar?

Theorem 1 can easily be generalized for the case when instead of a single R-action $\mathcal{L}$ we are given a set $\mathfrak{L}$ of R-actions and consider the following problem:

$\mathfrak{L}$-LIST REPLACEMENT TO A PLANAR GRAPH ($\mathfrak{L}$-LRP)
Input: A graph G and a non-negative integer k.
Question: Is there an R-action $\mathcal{L} \in \mathfrak{L}$ and $\varphi \in \mathbf{inj}([k], V(G))$ such that $\mathcal{L}_\varphi(G)$ is planar?

Since $|\mathcal{H}_k| = 2^{\binom{k}{2}}$, by brute force checking all R-actions of $\mathfrak{L}$ restricted to $\mathcal{H}_k$, we reduce this more general problem to $\mathcal{L}$-REPLACEMENT TO A PLANAR GRAPH and obtain the following corollary.

▶ **Corollary 2.** *For every family of R-actions $\mathfrak{L}$, there exists an algorithm that given an instance (G, k) of $\mathfrak{L}$-LIST REPLACEMENT TO A PLANAR GRAPH, reports whether (G, k) is a yes-instance in $O_k(|G|^2)$ steps.*

Another possibility to generalize the problem is to allow multiple actions, that is, to consider the following variant for a given R-action $\mathcal{L}$:

$\mathcal{L}$-CONSECUTIVE REPLACEMENT TO A PLANAR GRAPH ($\mathcal{L}$-CRP)
Input: A graph G, and two non-negative integers k and r.
Question: Is there a tuple $\varphi_1, \ldots, \varphi_h \in \mathbf{inj}([k], V(G))$ for some $h \leq r$ such that $\mathcal{L}_{\varphi_h}(\ldots \mathcal{L}_{\varphi_2}(\mathcal{L}_{\varphi_1}(G)) \ldots)$ is planar?

Notice that $\mathcal{L}_{\varphi_h}(\ldots \mathcal{L}_{\varphi_2}(\mathcal{L}_{\varphi_1}(G)) \ldots)$ can be seen as $\hat{\mathcal{L}}_\psi$ where $\psi \in \mathbf{inj}([k'], V(G))$ and $\hat{\mathcal{L}}$ is some R-action for $k' = |\varphi_1([k]) \cup \ldots \cup \varphi_h([k])|$. Since $k' \leq rk$, we can check all possible values of k' and for each k', construct the family of all possible R-actions $\mathfrak{L}$ that contain all feasible $\hat{\mathcal{L}}$ that can be results of compositions of $\mathcal{L}$. This way, we reduce $\mathcal{L}$-CONSECUTIVE REPLACEMENT TO A PLANAR GRAPH to $\mathfrak{L}$-LIST REPLACEMENT TO A PLANAR GRAPH and show that the problem is FPT when parameterized by $k + r$.

▶ **Corollary 3.** *For every R-action $\mathcal{L}$ and a positive integer r, there exists an algorithm that given an instance (G, k) of $\mathcal{L}$-CONSECUTIVE REPLACEMENT TO A PLANAR GRAPH, reports whether (G, k) is a yes-instance in $O_{k+r}(|G|^2)$ steps.*

In fact, it is possible to combine both generalizations and allow multiple actions chosen from a given list or even distinct lists.

3 High-level description of the algorithm

As our algorithm for $\mathcal{L}$-ARP is quite involved, in this extended abstract we present an outline of its main ideas. Our description is high-level. Formal definitions of the most important concepts in this description can be found in Section 4.

Basic concepts. We start with some conventions. In the course of our description, the word “small-enough” (resp. “big-enough” or “many-enough”) means upper (resp. lower) bounded by some function of k. By the term “flat part of G” we refer to some subgraph of G that can be embedded in a disk. Also by the term “bidimensionally big-enough part of a graph G” we refer to a flat part of G that contains a subdivision of a big-enough wall as a subgraph (see Figure 2). We say that a vertex of G is “well-enough insulated” if it is surrounded by a collection of many-enough homocentric cycles of some flat part of G. Intuitively, the existence of a bidimensionally big-enough part implies that a big-enough number of vertices of G are well-enough insulated.

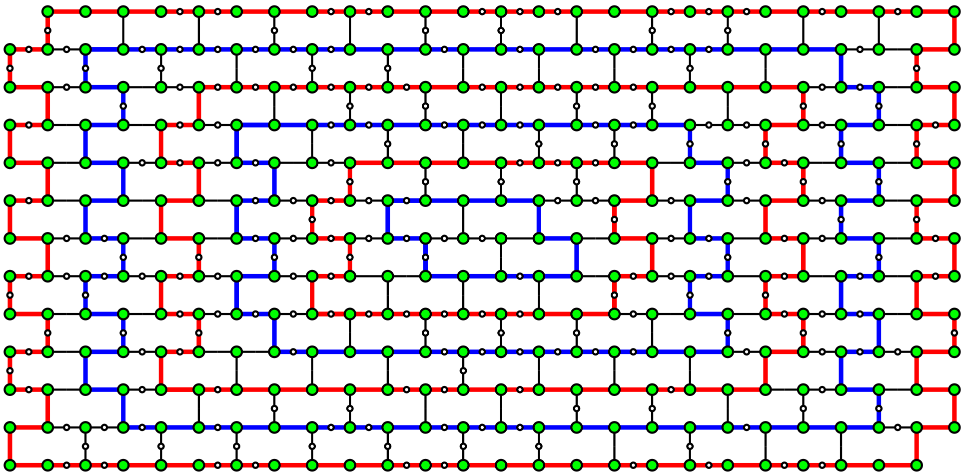

Figure 2 A subdivision of a 12×12-wall and the 6 homocentric cycles insulating its two central vertices.

Some preliminary observations. Before we present the main idea of the algorithm we start with some preliminary observations.

Observation 1: The kick-off observation is that it is possible to express $\mathcal{L}$-ARP in Monadic Second Order logic (MSOL). This implies that when the instance graph G has small-enough treewidth, then the problem can be solved in a linear in $n = |V(G)|$ number of steps. This observation is not straightforward as actions may also add edges in G and it requires some extra effort to translate this into MSOL.

Observation 2: The second observation is that if (G, k) is a yes-instance, then either G has bounded treewidth – and then we are done, because of Observation 1 – or it contains a bidimensionally big-enough flat part K. Moreover, using a result from [12, Subsection 4.1], we can also assume that K, besides the fact that it is bidimensionally big-enough, it has small-enough treewidth. This last property will permit us to apply MSOL-queries to any portion of K. It follows that K, if it exists, can be detected in $O(n)$ steps.

Observation 3: A third observation is that if some non-annotated vertex v in K is “surrounded” by a subdivided 3-wall-annulus A (see Figure 3) and the closed interior I_A of the outer cycle of A contains only non-annotated vertices, then we can remove v from G and reduce the instance to a simpler equivalent one. This reduction is based on the fact that the 3-connectivity of A offers enough rigidity for the graph inside I_A to remain unaffected in any planar graph that may be created by an action on G. Based on the above, we may assume that any bidimensionally big-enough territory of the flat part contains some annotated vertex in its interior.

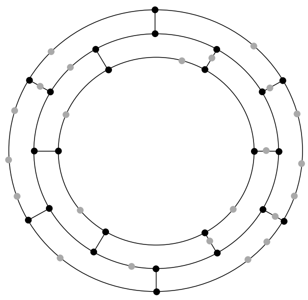

Figure 3 A subdivided 3-wall-annulus. The grey vertices are the subdivision vertices.

The territory-equivalence idea. The main idea of the algorithm is to detect, inside a bidimensionally big-enough part of G, a collection of many-enough pairwise-disjoint *territories* and to define a suitable notion of equivalence between them that expresses all the ways an action may affect them.

A critical aspect of our approach is that the number of equivalence classes of such an equivalence relation should depend *only* on k. This permits us, given that we have many-enough territories, to algorithmically detect some sub-collection of $k+1$ of them that are indistinguishable with respect to any action that can be applied on G. As an action affects vertices of at most k territories, we can arbitrary pick some particular annotated vertex inside one of them and create the equivalent instance where this vertex is no longer annotated. In this way, we can recurse to an equivalent instance of the problem that is more simple than the original one.

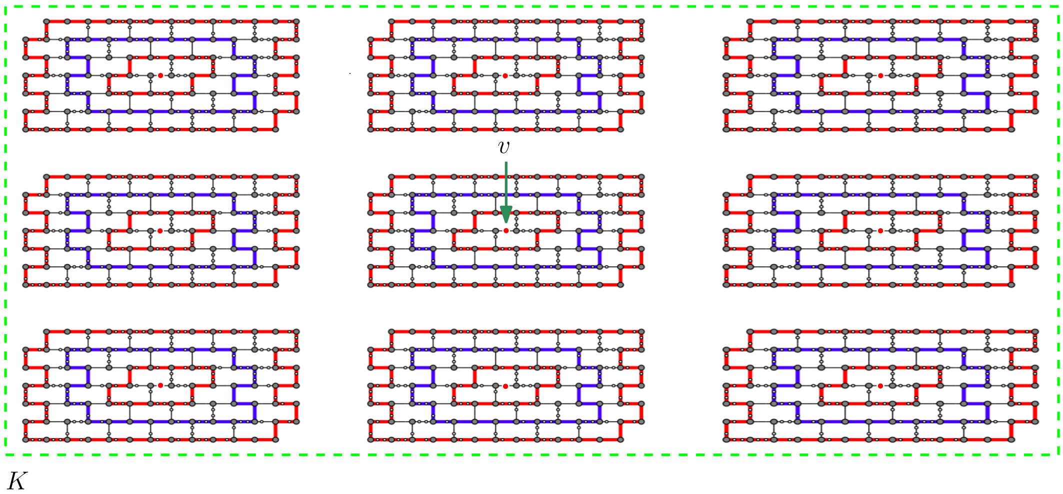

Figure 4 A visualization of the interior of the flat part K. Each subdivided sub-wall contains an annotated vertex v surrounded by many-enough homocentric cycles. In this particular figure, each subdivided sub-wall gives rise to a separation sequence of only 3 layers.

Separation sequences. We now explain how the above idea is implemented. First of all, we work on the flat part K (that can be detected due to the Observation 2) and use the fact that it is bidimensionally big-enough in order to detect in it a big-enough set X of well-enough insulated annotated "central" vertices. Thus each $v \in X$ is accompanied with a big-enough collection $\mathcal{C}_v = \{C_1, \ldots, C_r\}$ of surrounding cycles, called *a separation sequence* (assuming that C_1 is the outermost cycle). Moreover, based on the big-enough bidimensionality of K we can also assume that for every two distinct $v, v' \in X$, the closed interiors of the outermost cycles of $\mathcal{C}_v$ and $\mathcal{C}_{v'}$ are disjoint (see Figure 4). Based on Observations 2 and 3, it is possible to detect such a set X of annotated vertices along with their accompanying cycle collections, in $O(n)$ steps. Moreover, for each cycle C_j in $\mathcal{C}_v$ we denote by G_j the graph cropped by its interior and we make sure that the 3 outermost cycles are parts of a subdivided 3-wall-annulus A_j. We call G_j *j-th prefix graph of v*.

We see each prefix graph as a doubly annotated graph $\mathbf{G}_j = (G_j, Z_j, R_j)$ where Z_j are the vertices inside the annulus A_j and R_j are the annotated vertices inside G_j. As r is big-enough, we know that for any application of an action on a set S of k vertices, that is able to transform G to a planar graph, there will be some $Z_j, j \in \{1, \ldots, r\}$ that will be disjoint from S. This permits us to see this Z_j as a *separator* of G whose inner part (including Z_j) is G_j. Moreover after an action that does not affect Z_j, the set Z_j will maintain its status as a separator in the resulting graph. This notion of separator is formalized by the concept of an *annulus-embedded separator* that is formally described in Subsection 4.2.

An equivalence relation on prefix graphs. Recall that a prefix graph G_j is a planar graph of bounded treewidth. We stress that typical equivalence relations that are based on MSOL-expressible properties require that the boundary of the bounded-treewidth graphs is also bounded. However, this is not our case. Instead we will use the fact that the boundary Z_j "well insulates" G_j from the rest of the graph.

We need to encode, for each $\mathbf{G}_j = (G_j, Z_j, R_j)$, all possible ways an action may rearrange G in both sides of the separator Z_j, assuming that it does not affect the vertices of Z_j. Clearly, if S is the set of vertices of G on which an action is applied, then some part of S is outside G_j and another part is inside G_j (but not in Z_j). As the action does not affect the separator Z_j, it will create a planar graph G' where the outer and inner part will still be well-defined with respect to Z_j.

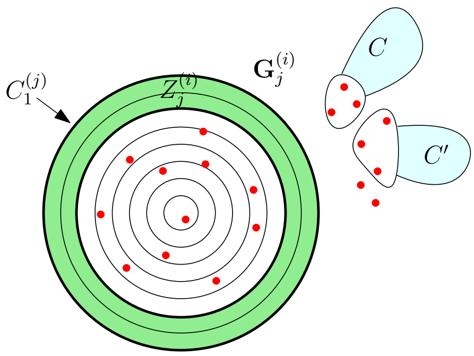

Figure 5 A prefix graph G_i. The set Z_j are the vertices in the annulus defined by its 3 first layers. The red vertices are the vertices affected by some action. The graphs C' and C are connected components of $(G \setminus V(G_j)) \setminus S$ that will be relocated after the application of an action to G.

The vertices of S along with the connected components of $(G \setminus V(G_j)) \setminus S$ will be relocated in G' in the inner and the outer parts of Z_j. However, these possible relocations depend only one the size of S (that is k) and the ways these components can be seen as "partially-planar graphs" with respect to their boundary in S (which also depends on k). The precise encoding of this is expressed by the notion of a *replacement folio of a doubly annotated graph*, formally defined in Subsection 4.3. This encoding is justified by a Lemma asserting (the proof is omitted in this extended abstract) that any action on G that avoids Z_j can be "represented" by this folio in the sense that if two doubly annotated graphs, that correspond to different separation sequences $\mathcal{C}_v$ and $\mathcal{C}_{v'}$, have the same replacement folio, then the application of an action to the first can be simulated by the application of an action to the second.

An important technical step is to prove a series of lemmata that "translate" all possible elements of a replacement folio to MSOL-formulas. Again we stress that this is not straightforward: besides the need of suitably encoding partially-planar graphs, we also have to express all the ways "missing edges" of G may appear as the result of an action. As a consequence of its MSOL-expressibility, it follows that our equivalence relation on territories has a small-enough number of equivalence relations. This, together with the bounded treewidth of the flat part K, permits us to detect in it $k+1$ equivalent territories in $O_k(n)$ steps, "de-annotate" one of their annotated "central vertices" and create a simpler equivalence instance. Given this reduction and the reduction of Observation 3, we may use $|V(G)|+|R| \leq 2|V(G)|$ as the complexity-measure of an instance of our algorithm. Therefore, after $O(n)$ recursive calls of the above reductions, the algorithm provides a correct answer or an equivalent instance whose graph has small-enough treewidth. In the latter case, the correct answer can be computed in $O_k(n)$ steps, as mentioned in Observation 1. Based on the above, the overall running time of the algorithm is $O_k(n^2)$.

4 Key concepts

4.1 Graph embeddings and boundaried graphs

Embedded graphs. We denote by $\mathbb{S}_0$ the sphere $\{(x, y, z) \mid x^2 + y^2 + z^2 = 1\}$ and we refer to it as *the plane*. We consider embeddings or partial embeddings of graphs on $\mathbb{S}_0$ and several subsets of it. Such subsets can be closed *disks*, i.e. subsets of $\mathbb{S}_0$ that are homeomorphic to the set $\{(x, y, z) \mid x^2+y^2 \leq 1\}$ and *closed annuli*, i.e., subsets of $\mathbb{S}_0$ that are homeomorphic to the set $\{(x, y, z) \mid \frac{1}{2} \leq x^2 + y^2 \leq 1\}$. Given a set X that is either a closed disk or an annulus, we denote its boundary (i.e., the set of points of X for which every neighborhood around them contains some point not in Z) by $\mathbf{bor}(X)$. Notice that if A is an annulus, then the set $\mathbf{bor}(A)$ has two connected components that are both cycles. We call these cycles *boundaries* of A. An *oriented annulus* is a triple $\mathbb{A} = (A, N_{\text{in}}, N_{\text{out}})$ where A is an annulus and N_{in} and N_{out} are its boundaries. We say that N_{in} (resp. N_{out}) is the *inner* (resp. *outer*) boundary of $\mathbb{A}$. Given an oriented annulus $\mathbb{A} = (A, N_{\text{in}}, N_{\text{out}})$ we define $\mathbf{rev}(\mathbb{A}) = (A, N_{\text{out}}, N_{\text{in}})$, i.e., we exchange the roles of the inner and the outer boundary. When we embed a graph G in the plane, in the annulus, or in a disk, we treat G as a set of points. This permits us to make set operations operations between graphs and sets of points. For instance, if G is a graph embedded in the plane and Δ is a closed disk in the plane, we can use the notation $V(G) \cap D$ in order to the set of vertices of G that are points of D. Also, given that $\mathbf{bor}(D) \cap G \subseteq V(G)$, we use $G \cap D$ to denote the graph formed by the vertices and the edges of G that are inside D. We denote by $\mathcal{P}$ the class of all planar graphs.

Annotated graphs. An *annotated graph* is a pair (G, R) where G is a graph and $R \subseteq V(G)$. A triple (G, R, Z) where $R, Z \subseteq V(G)$ is called *doubly annotated graph*.

Boundaried graphs. Let $k \in \mathbb{N}$. A *k-boundaried graph* is a triple $\mathbf{G} = (G, B, \lambda)$ where (G, B) is an annotated graph and $\lambda \in \mathbf{inj}(B, [k])$ (keep in mind that $|B| \leq k$). For every $x \in R$, we refer to the number $\lambda(x)$ as the *index* of x in $\mathbf{G}$ and we define the *index set* of $\mathbf{G}$ as $\Lambda(\mathbf{G}) = \lambda(B)$. We call B the *boundary* of $\mathbf{G}$ and the vertices of B *the boundary vertices* of $\mathbf{G}$. We also denoted $B(\mathbf{G}) = B$ and $\lambda(\mathbf{B}) = \lambda$. Also we define the *size*, denoted by $|\mathbf{G}|$ of $\mathbf{G} = (G, B, \lambda)$ by $|G|$. We denote by $\mathcal{B}_k$ the set of all k-boundaried graphs, for each $k \in \mathbb{N}$. Given a $\mathbf{G} = (G, B, \lambda) \in \mathcal{B}_k$ we define the graph $\mathsf{gr}(\mathbf{G}) = (\Lambda(\mathbf{G}), \{\lambda(e) \mid e \in E(G[B])\})$, i.e., $\mathsf{gr}(G)$ is the graph in $\mathcal{H}_k$ that is obtained after taking the subgraph of G induced by the boundary $B(\mathbf{G})$ and then mapping it, via λ, to numbers in $[k]$. Also, given two $i, j \in \Lambda(G)$, we say that $i \sim_{\mathbf{G}} j$ if $\lambda^{-1}(i)$ and $\lambda^{-1}(j)$ belong in the same connected component of G. Clearly, $\sim_{\mathbf{G}}$ is an equivalence relation. Let $\mathbf{G}_1, \mathbf{G}_2 \in \mathcal{B}_k$. We say that $\mathbf{G}_1$ and $\mathbf{G}_2$ are *compatible* if $\mathsf{gr}(\mathbf{G}_1) = \mathsf{gr}(\mathbf{G}_2)$ and $\sim_{\mathbf{G}_1} = \sim_{\mathbf{G}_2}$.

Let $\mathbf{G}_1 = (G_1, B_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, B_2, \lambda_2)$ be two compatible k-boundaried graphs. We define the *gluing operation* $\oplus$ such that $\mathbf{G}_1 \oplus \mathbf{G}_2$ is the graph obtained by taking the disjoint union of G_1 and G_2 and then identifying each vertex in B_1 with the same-indexed vertex in B_2. We make the convention that after identifying a vertex $x \in B_1$ with a vertex $y \in B_2$, then the result of this identification in $\mathbf{G}_1 \oplus \mathbf{G}_2$ is again the vertex x (i.e., B_1 prevails over B_2). Finally, we say that $\mathbf{G}_1 \equiv \mathbf{G}_2$, if they are compatible and for every boundary graph $\mathbf{F}$ where $\mathsf{gr}(\mathbf{F}) = \mathsf{gr}(\mathbf{G}_i), i \in [2]$ it holds that

$$\mathbf{F} \oplus \mathbf{G}_1 \in \mathcal{P} \iff \mathbf{F} \oplus \mathbf{G}_2 \in \mathcal{P}. \tag{1}$$

4.2 Annulus-embedded separators

The notion of a subdivided 3-wall-annulus is depicted in Figure 3. Notice that each such graph contains two "boundary" cycles that we call *extremal cycles*.

Given a $S \subseteq V(G)$, we define as $\mathbf{ccin}(G, S)$ as the set of all connected components of $G \setminus S$ that are not connected components of G. We also define $\mathbf{ccout}(G, S)$ as the union of all connected components of $G \setminus S$ that are connected components of G.

Annulus-boundaried graphs. An *annulus-boundaried graph* is a quadruple $(C, K, Y, \mathbb{A})$ where

- C is a graph,
- K is a connected subgraph of C,
- Y is a subdivided 3-wall-annulus that is a subgraph of K,
- $\mathbb{A} = (A, N_{\text{in}}, N_{\text{out}})$ is an oriented annulus,
- Y is embedded in A so that
  - N_{in} and N_{out} are the two extremal cycles of Y, and
  - $G \cap A = K$.

We call the cycle of Y that is identical to N_{in} (resp. N_{out}) *inner* (resp. *outer*) cycle of $(C, K, Y, \mathbb{A})$. We say that an annulus-boundaried graph $(C, K, Y, \mathbb{A})$ is *planar* if Q can be embedded in a disk Δ such that $\mathbb{A} \subseteq \Delta$ and $N_{\text{out}} = \mathbf{bor}(\Delta)$.

Annulus-embedded separators. Let G be a graph. Let also $(K, Y, \mathbb{A})$ be a triple where K is a graph, Y is a subgraph of K and $\mathbb{A}$ is an oriented annulus. We say that $(K, Y, \mathbb{A})$, is a *annulus-embedded separator* of G if there are two subgraphs C_{in} and C_{out} of G such that both $(C_{\text{in}}, K, Y, \mathbb{A})$ and $(C_{\text{out}}, K, Y, \mathbf{rev}(\mathbb{A}))$ are annulus-boundaried graphs. Notice that each connected component of $\mathbf{ccin}(G, V(K))$ contains some vertex x where $N_G(x)$ intersects either the inner or the outer cycle of $(G, K, Y, \mathbb{A})$ (but not both). We also make the convention that all connected components of $\mathbf{ccout}(G, V(K))$ are subgraphs of C_{out} and we denote C_{in} (resp. C_{out}) *inner* (resp. *outer) component* of $(K, Y, \mathbb{A})$ in G. We say that $(K, Y, \mathbb{A})$ is *inner-planar* if $(C_{\text{in}}, K, Y, \mathbb{A})$ is planar.

The following observation easily directly from the definitions and the fact that that every 3-sd-annulus is 3-connected and has a unique embedding in the plane.

▶ **Observation 4.** *Let (G, R) be an annotated graph and let $(K, Y, \mathbb{A})$ be a annulus-embedded separator of G. Let also C_{in} and C_{out} be the inner and the outer component of G respectively. Then G is planar if and only if both C_{in} and C_{out} are planar.*

4.3 Replacement folios

A *k-planarity-folio* is a set $\mathcal{M}_k \subseteq \mathcal{B}_k$ such that for every $\mathbf{G} \in \mathcal{B}_k$ there is a $\mathbf{J} \in \mathcal{M}_k$ such that $\mathbf{J} \equiv \mathbf{G}$. It is known (see e.g., in [2]) that, for every $k \in \mathbb{N}$, it is possible to construct a k-planarity-folio $\mathcal{M}_k$ whose size depends on k. Given a $k \in \mathbb{N}$, we define $g_k = \max\{|\mathbf{C}| \mid \mathbf{C} \in \mathcal{M}_k\}$. For every graph L where $V(L) \subseteq [k]$, we define $\mathcal{C}_L = \{\mathbf{G} \in \mathcal{M}_k \mid \mathsf{gr}(\mathbf{G}) = L\}$. The following is an direct consequence of the definitions (see e.g., [2]).

▶ **Observation 5.** *For every k-numbered graph L and every two compatible boundaried graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ where $\mathsf{gr}(\mathbf{G}_1) = L$, it holds that $\mathbf{G}_1 \equiv \mathbf{G}_2$ if and only if $\forall \mathbf{F} \in \mathcal{C}_L$, $\mathbf{F} \oplus \mathbf{G}_1 \in \mathcal{P} \iff \mathbf{F} \oplus \mathbf{G}_2 \in \mathcal{P}$.*

From now on we fix some k-planarity folio $\mathcal{M}_k$. Given the above observation, the members of the $\mathcal{M}_k$ represent all ways a boundary graph can be "partially planar" with respect to its boundary.

Replacement folios. Our purpose now is to define a structure representing the effect of all replacement actions on a doubly annotated graph (G, R, Z). The actions we want to encode involve vertices in R that are not in Z. Also they involve vertices of some virtual graph D that is not a part of G. Later, the set Z will be the vertex set of a wall embedded separator of a graph and D will represent the part of the graph, affected by an action, that is in the outer component of this embedded separator.

Let $k \in \mathbb{N}$. A *k-pattern* is a quadruple $(\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}}, D, \tau)$ where

- $\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}} \in \mathcal{M}_k$ (both $\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}}$ are members of the k-planarity folio),
- $\Lambda(\hat{\mathbf{J}}_{\text{in}}) \cap \Lambda(\hat{\mathbf{J}}_{\text{out}}) = \emptyset$ ($\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}}$ do not have same-index boundary vertices),
- D is a graph where $|D| = |\mathsf{gr}(\hat{\mathbf{J}}_{\text{in}}) \cup \mathsf{gr}(\hat{\mathbf{J}}_{\text{out}})|$ (the size of D is the sum of the boundaried sizes of $\hat{\mathbf{J}}_{\text{in}}$, and $\hat{\mathbf{J}}_{\text{out}}$), and
- τ is an bijection from $V(D)$ to $\Lambda(\mathbf{J}_{\text{in}}) \cup \Lambda(\mathbf{J}_{\text{out}}))$ (i.e., τ is a labelling of D with numbers from the index sets of $\mathbf{J}_1$ and $\mathbf{J}_2$).

We denote by $\mathcal{P}_k$ the set of all k-patterns. Suppose now that (G, R, Z) is a doubly annotated graph. The *k-planar $\mathcal{L}$-replacement folio* of (G, R, Z) is the set $\mathfrak{L}_k(G, R, Z) \subseteq \mathcal{P}_k$ containing every quadruple $(\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}}, D, \tau) \in \mathcal{P}_k$ for which there exists an $\varphi \in \mathbf{inj}([k], V(D) \cup (R \setminus Z))$ such that

1. $\tau^{-1} \subseteq \varphi$,
2. if $G' = \mathcal{L}_\varphi(G \cup D)$, then $\tau(G') = \mathsf{gr}(\hat{\mathbf{J}}_{\text{in}}) \cup \mathsf{gr}(\hat{\mathbf{J}}_{\text{out}})$

and if

- $S_{\text{in}} = \varphi(\Lambda(\hat{\mathbf{J}}_{\text{in}}))$,
- $S_{\text{out}} = \varphi(\Lambda(\hat{\mathbf{J}}_{\text{out}}))$,
- $\hat{G} = (G', S_{\text{in}}, \varphi^{-1}|_{S_{\text{in}}}) \oplus \hat{\mathbf{J}}_{\text{in}}$,
- $\hat{U}_{\text{in}} = \mathbf{ccin}(\hat{G}, Z)$, and
- $\hat{U}_{\text{out}} = \mathbf{ccout}(\hat{G}, Z)$,

then

3. $\hat{U}_{\text{in}}$ is planar,
4. $S_{\text{out}} \subseteq V(\hat{U}_{\text{out}})$, and
5. $(\hat{U}_{\text{out}}, S_{\text{out}}, \varphi^{-1}|_{S_{\text{out}}}) \equiv \hat{\mathbf{J}}_{\text{out}}$.

Some intuition. Let us give some intuition on the above, quite technical, definition of the set $\mathfrak{L}_k(G, R, Z)$. Suppose that $(\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}}, D, \tau) \in \mathcal{P}_k$. The graph G should be seen as an annulus-boundaried graph $(G, K, Y, \mathbb{A})$ where $Z = V(K)$ where $(K, Y, \mathbb{A})$ is an annulus-embedded separator between G and a virtual "outside part". Assume also that φ is an action that affects vertices in $R \setminus Z$ (i.e., vertices in the interior, call it I_G, of the wall separator, but not in R) and some virtual vertices outside G. The graph D represents these vertices and the way these are connected between them (see Figure 6).

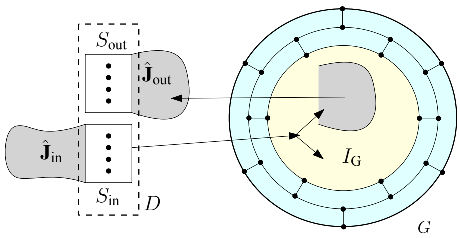

Figure 6 A 3-sd-annulus. The grey vertices are the subdivision vertices.

Notice that the action φ creates some edges between $V(D)$ and I_G. Moreover, the same action is rearranging the edges in D and the edges between the vertices in $\varphi([k]) \cap I_G$. The edges of D are rearranged so as to transform it to a graph isomorphic to $\mathsf{gr}(\hat{\mathbf{J}}_{\text{in}}) \cup \mathsf{gr}(\hat{\mathbf{J}}_{\text{out}})$ and this is enforced by conditions 1 and 2. Let G' be the result of the application of φ on $G \cup D$ and the $\hat{G}$ be result of gluing G' with $\hat{\mathbf{J}}_{\text{in}}$. Here $\mathbf{J}_{\text{in}}$ represents a virtual "outside part" that is not present in G. Notice that the connected components of this outside part are finally scattered either inside or outside the resulting graph. Moreover both parts should be finally "boundaried parts" of a planar graph and this is the reason we incorporated both planarity and connectivity in the definition of the k-planarity folio. Now $\hat{U}_{\text{in}}$ is the part of $\hat{G}$ that goes "inside" and $\hat{U}_{\text{out}}$ is the part of $\hat{G}$ that goes "outside" after the rearrangement. In condition 3 we demand the inside part $\hat{U}_{\text{in}}$ to be planar (we stress that this "inside part" may contain connected components of $\hat{\mathbf{J}}_{\text{in}}$). We also demand that outside part $\hat{U}_{\text{out}}$ contains S_{out} (condition 4) and that $\hat{U}_{\text{out}}$ is boundaried by S_{out} is equivalent to $\hat{\mathbf{J}}_{\text{out}}$ because of Condition 5 (again it is possible that this boundaried graph contains connected components of $\hat{\mathbf{J}}_{\text{in}}$). In this way, $\hat{\mathbf{J}}_{\text{out}}$ represents the virtual "outside part" the separator. Resuming, if $(\hat{\mathbf{J}}_{\text{in}}, \hat{\mathbf{J}}_{\text{out}}, D, \tau) \in \mathcal{P}_k$ then there is an action φ that outside G behaves as indicated by D and τ, that assumes that the virtual outside part is equivalent to $\hat{\mathbf{J}}_{\in}$ and, under this assumption, demands that the occurring outside part is equivalent to $\hat{\mathbf{J}}_{\text{out}}$.

As already mentioned in the high-level description of our algorithm, all conditions of the above definition can be expressed in Monadic Second Order logic. We omit the proof of this fact in this extended abstract.

5 Applications

5.1 Problems generated by different Instantiations of $\mathcal{L}$

$\mathcal{L}$-ARP can express several modification operations based on different instantiations of $\mathcal{L}$. In this section we give a series of examples. We slightly extend the definition of an action by demanding that $\mathcal{L} : \mathcal{H} \to \mathcal{H}$ is a *partial* function, i.e., we allow that $\mathcal{L}(H')$ may be undefined for some $H' \in \mathcal{H}$ and, in such a case, we write $\mathcal{L}(H') = \mathsf{void}$. Moreover, in the question of $\mathcal{L}$-ARP we additionally demand that $\mathcal{L}(\varphi^{-1}(G)) \neq \mathsf{void}$. To reduce the enhanced version to the old one, we define G' as the disjoint union of G and K_5, we set $k' = k + 5$ and set up a $\mathcal{L}'$ so that $\mathcal{L}'(H)$ is defined as follows: if H' is the disjoint union of a graph H and a K_5 and $\mathcal{L}(H) \neq \mathsf{void}$, then $\mathcal{L}'(H')$ is equal to the disjoint union of $\mathcal{L}(H)$ and K_5^- (that is the graph K_5 without one edge, i.e., a planar graph), otherwise $\mathcal{L}(H) = H$. We now observe that (G, k) is a yes-instance of the enhanced $\mathcal{L}$-ARP iff (G', k') is a yes-instance of $\mathcal{L}'$-ARP.

We now proceed with a series of problems that can be easily expressed by the enhanced version of $\mathcal{L}$-ARP.

Planar Completion to a Subgraph. This problem has as input two planar graphs G and H and asks whether it is possible to add edges in G so that the resulting graph remains planar (a *planar completion* of G) and contains H as a subgraph. A variant of this problem, where G is given along with some plane embedding, has been examined in [4]. We set $k = |H|$. By setting $\mathcal{L} : \mathcal{H} \to \mathcal{H}$ where $\mathcal{L}(H') = H$, i.e., $\mathcal{L}$ is the constant function where the output is always H, we reduce Edge Completion to Subgraph to $\mathcal{L}$-ARP.

Planar Completion to an Induced Subgraph. Here, given G and H, we ask for a planar completion of G that contains H as an induced subgraph. To reduce this problem to $\mathcal{L}$-ARP, we consider $\mathcal{L} : \mathcal{H} \to \mathcal{H}$ where $\mathcal{L}(H') = H$ if H' is an induced subgraph of H, otherwise $\mathcal{L}(H') = \mathsf{void}$.

Edge Deletion to a Planar Graph. Edge Deletion to a Planar Graph asks whether we can remove at most k edges from G such that the resulting graph will be planar. A pair $(H, F) \in \mathcal{H}_{2k} \times \mathcal{H}_{2k}$ is *good* if $E(F) \subseteq E(H)$ and $|E(H) \setminus E(F)| \leq k$. For every good pair (H, F), we define the action $\mathcal{L}_{H,F} : \mathcal{H} \to \mathcal{H}$ where $\mathcal{L}_{H,F}(H') = F$ if $H' = H$, otherwise we set $\mathcal{L}_{H,F}(H') = \mathsf{void}$. Notice that (G, k) is a yes instance of Edge Deletion to a Planar Graph iff there is a good pair (H, F) where (G, k) is a yes-instance of $\mathcal{L}_{H,F}$-ARP. As there are $O_k(1)$ good pairs, this implies an FPT-algorithm for Edge Deletion to a Planar Graph when parameterized by k.

Alternativelly, we can see this problem as a special version of $\mathcal{L}$-CRP by exchanging the roles of r and k, setting $k = 2$, $\mathcal{L}(K_2) = \overline{K}_2$, and $\mathcal{L}(\overline{K}_2) = \overline{K}_2$, where $\overline{K}_2$ is the complement of the complete graph on two vertices.

Matching Deletion to a Planar Graph. We ask here whether we can remove a matching of size at most k in order to create a planar graph. The reduction is again the same as in the case of Edge Deletion to a Planar Graph, however now, for a pair $(H, F) \in \mathcal{H}_{2k} \times \mathcal{H}_{2k}$ to be good, we additionally ask that $E(H) \setminus E(G)$ is a matching of H.

Planar Subgraph Isomorphism. Planar Subgraph Isomorphism has as input two planar graphs G and J and asks whether G contains J as a subgraph. We define the action $\mathcal{L} : \mathcal{H} \to \mathcal{H}$ where $\mathcal{L}(H') = H'$ if J is a subgraph of H', otherwise $\mathcal{L}(H') = \mathsf{void}$. Then $\mathcal{L}$-ARP is the Planar Subgraph Isomorphism.

Planar Induced Subgraph Isomorphism. Planar Induced Subgraph Isomorphism has as input two planar graphs G and J and asks whether G contains J as an induced subgraph. The construction of $\mathcal{L}$ is as in the previous case with the difference that we now demand that J is isomorphic to H'.

Edge-disjoint Planar Superposition: given two planar graphs G and H, check whether H is a subgraph of the complement of G and $H \cup G$ is planar. We define the action $\mathcal{L} : \mathcal{H} \to \mathcal{H}$ where $\mathcal{L}(H') = H$ if $E(H') = \emptyset$, otherwise $\mathcal{L}(H') = \mathsf{void}$. Then $\mathcal{L}$-ARP is the EDGE-DISJOINT PLANAR SUPERPOSITION.

5.2 Modifications to planar graph with additional properties

Let $\mathcal{G}$ be a graph property, i.e, a subset of the set of all graphs. We consider the following extension of $\mathcal{L}$-RP.

$\mathcal{L}$-REPLACEMENT TO A PLANAR GRAPH WITH PROPERTY $\mathcal{G}$ ($\mathcal{L}$-RPP($\mathcal{G}$)).
Input: A graph G and a non-negative integer k.
Question: is there a $\varphi \in \mathbf{inj}([k], V(G))$, such that $\mathcal{L}_\varphi(G)$ a planar graph in $\mathcal{G}$?

We now provide some instantiations of $\mathcal{G}$ for which the $\mathcal{L}$-RPP($\mathcal{G}$) belongs in FPT, when parameterized by k. In each case we explain how our algorithm should be modified.

(1) $\mathcal{G} := \mathcal{G}_H$ is the set of all H-subgraph-free graphs, for some connected graph H. In the definitions of an annulus-boundaried graph and wall embedded separators (see Subsection 4.2) instead of taking a 3-sd-wall, we now consider an r-sd-wall where r is the diameter of H. This permits the modification of the conclusion of Observation 4 to "G is H-subgraph free and planar if and only if both C_{in} and C_{out} are H-subgraph free and planar". Also we may enhance the definition of a replacement folio (defined formally in Subsection 4.3) by considering in Condition 3 $\hat{U}_{\text{in}}$ to be planar and H-subgraph free and in Condition 5, we use a revised version of the equivalence $\equiv$ where we instead demand in (1) that $\mathbf{F} \oplus \mathbf{G}_1 \in \mathcal{P} \cap \mathcal{G} \iff \mathbf{F} \oplus \mathbf{G}_2 \in \mathcal{P} \cap \mathcal{G}$. Notice that for every H, $\mathcal{G}_H$ is a MSOL-expressible property (actually it is even expressible in First Order Logic) therefore the revised $\equiv$ also has finite index.

(2) $\mathcal{G} = \mathcal{G}_H$ is the set of all induced H-subgraph-free graphs, for some connected graph H. The modifications are exactly the same as in the previous case. Just take into account that H-induced minor freeness is MSOL-expressible.

(3) $\mathcal{G} := \mathcal{G}_d$ is the set of all d-regular graphs for $d \in \{3, 4, 5\}$. In this case, one can easily verify that the following relaxed version of the conclusion of Observation 4 holds: "G is d-regular and planar if and only if both C_{in} and C_{out} are almost d-regular and planar". Here by "almost" we demand the vertices of the 3-sd wall $\mathbb{A} = (A, N_{\text{in}}, N_{\text{out}})$ that are on N_{out} to have degree at most d and all the others to have degree exactly d. Notice that this relaxed regularity condition is again expressible in MSOL-logic (for this we need to annotate the vertices in N_{out}). As in the previous cases, we can again demand, in Condition 3, that $\hat{U}_{\text{in}}$ is planar and almost 3-regular and, in Condition 5 enhance the definition of the equivalence relation by additionally asking almost regularity in (1).

(4) $\mathcal{G}$ is the set of all Eulerian graphs. This is the same as in the previous case. The only difference is that we now ask degrees to be even. All modifications are are parallel to the previous case (notice that connectivity demand for Eulerian graphs is already incorporated in the definition of $\equiv$). However, asking for a graph to have all vertices, except possibly from some annotated set, of even degree is not MSOL-expressible. However, we can use an extension of MSOL, called *Counting MSOL* (CMSOL) that is MSOL with an additional predicate $\mathbf{card}_{q,r}$ where $(G, S) \models \mathbf{card}_{q,r} \iff |S| \equiv q \pmod{r}$. It is known (see e.g. [3, Lemma 3.2]) that equivalence relations on CMSOL-expressible properties have finite index. This permits us to deal with the parity demand of being Eulerian.

(5) $\mathcal{G}$ is the set of all bipartite graphs. As before, one can easily derive a version of Observation 4 where the conclusion is: "G is planar bipartite if and only if both C_{in} and C_{out} are planar bipartite'. The modifications are analogous to those of the previous cases where the property is the exclusion of an odd cycle, which is CMSOL-expressible.

(6) $\mathcal{G}$ is the set of all triangulated graphs. Notice that $\mathcal{G}$ contains every graph G that has exactly $3|G| - 6$ edges. Let (G, k) be an instance of $\mathcal{L}$-RPP($\mathcal{G}$) and let $t_G = |E(G)| - (3|V(G)| - 6)$. As an action cannot remove or introduce more than $\binom{k}{2}$ edges, we assume that $-\binom{k}{2} \leq t_G \leq \binom{k}{2}$, otherwise (G, k) is a no-instance. Under this demand, $\mathcal{L}$ should change to $\mathcal{L}'$ where $\mathcal{L}'(H) = \mathcal{L}(H)$ if $|E(H)| - |E(\mathcal{L}(H))| = t_G$ and $\mathcal{L}'(H) = \mathsf{void}$ otherwise. As the action $\mathcal{L}''$ when applied on G always creates a graph G' that has $3|G'| - 6$ edges and in the case G' is planar, then it should also be triangulated. Therefore (G, k) is a yes instance of $\mathcal{L}$-ARP($\mathcal{G}$) iff (G', k) is a yes instance of $\mathcal{L}'$-ARP.

(7) $\mathcal{G}$ is the set of all radial graphs. A radial graph is a graph that can be embedded in the plane so that all its faces are squares. Recall that a graph G is radial if it is planar, bipartite, and has $2|G| - 4$ edges. We apply the same reduction as in the previous case for $t_G = |E(G)| - (2|V(G)| - 4)$ and we have that $\mathcal{L}$-RPP($\mathcal{G}$) iff (G', k) is a yes instance of $\mathcal{L}'$-ARP($\mathcal{G}'$) where $\mathcal{G}'$ is the class of all bipartite graphs (treated in (5)).

6 Conclusions and open problems

In this paper we proved that for every editing operation that is based on adjacency modifications, the planarization problem is FPT, when parameterized by the number of edges that are changed during this modification. We have also seen that the formalization of modification problem by actions is quite versatile and can express most known modifications problems of this flavour. The are three possible extensions of our results that could induce further research on this topic.

First one may consider more general modifications where some (bounded) part of the graph is replaced by another, however not of the same size but still bounded. We believe that this setting is amenable to the techniques that we introduce in this paper, however more complicated to deal with. Also, one may consider modifications that involve the whole (unbounded) neighbourhood of a bounded part of the graph. Contractions or vertex removals would fit in such a more general framework.

Second one may consider, instead of planar graphs, other classes of graphs such as graphs of bounded genus or graphs excluding a graph as a minor, where one may still employ structural results about "flat territories". We believe that our machinery can be extended in this direction. However, such extensions should be quite non-trivial as they involve several technicalities on how separators may split graphs in those families and how actions may rearrange them.

A third direction is to find more examples of, additional to planarity, target properties. Most of the examples in Subsection 5.2 demand changes either to the way Observation 4 applies or to the equivalence relation $\equiv$ that may still keep it of finite index. Is there a way to systematize this into a general meta-algorithmic framework?

References

1 Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Tight Bounds for Linkages in Planar Graphs. In *Proceedings of the 38th International Colloquium of Automata, Languages and Programming (ICALP)*, volume 6755 of *Lecture Notes in Comput. Sci.*, pages 110–121. Springer, 2011.

2 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Optimal Algorithms for Hitting (Topological) Minors on Graphs of Bounded Treewidth. In *12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria*, pages 4:1–4:12, 2017.

3 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) Kernelization. *J. ACM*, 63(5):44:1–44:69, 2016.

4 Dimitris Chatzidimitriou, Archontia C. Giannopoulou, Spyridon Maniatis, Clément Requilé, Dimitrios M. Thilikos, and Dimitris Zoros. FPT algorithms for plane completion problems. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 26:1–26:13, 2016.

5 Bruno Courcelle. The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic. *Handbook of Graph Grammars*, pages 313–400, 1997.

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

7 Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. The planar directed k-Vertex-Disjoint Paths problem is fixed-parameter tractable. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 197–207. IEEE, 2013.

8 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting Forbidden Minors: Approximation and Kernelization. *SIAM Journal on Discrete Mathematics*, 30(1):383–410, 2016. doi:10.1137/140997889.

9 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 470–479. IEEE, 2012.

10 Archontia C Giannopoulou, Bart MP Jansen, Daniel Lokshtanov, and Saket Saurabh. Uniform kernelization complexity of hitting forbidden minors. *ACM Transactions on Algorithms*, 13(3):35, 2017.

11 Archontia C Giannopoulou, Michał Pilipczuk, Dimitrios M Thilikos, Jean-Florent Raymond, and Marcin Wrochna. Linear kernels for edge deletion problems to immersion-closed graph classes. *arXiv preprint arXiv:1609.07780*, 2016.

12 Petr A. Golovach, Marcin Kaminski, Spyridon Maniatis, and Dimitrios M. Thilikos. The Parameterized Complexity of Graph Cyclability. *SIAM J. Discrete Math.*, 31(1):511–541, 2017. doi:10.1137/141000014.

13 Petr A. Golovach, Pim van 't Hof, and Daniël Paulusma. Obtaining planarity by contracting few edges. *Theor. Comput. Sci.*, 476:38–46, 2013. doi:10.1016/j.tcs.2012.12.041.

14 Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 479–488. ACM, 2011.

15 Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A Near-optimal Planarization Algorithm. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1802–1811. SIAM, 2014. URL: http://dl.acm.org/citation.cfm?id=2634074.2634204.

16 Ken-ichi Kawarabayashi. Planarity Allowing Few Error Vertices in Linear Time. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–648. IEEE, 2009.

17 Ken-ichi Kawarabayashi and Yusuke Kobayashi. The induced disjoint path problem. In *13th Conference on Integer Programming and Combinatorial Optimization, IPCO 2008*, volume 5035 of *Lecture Notes in Computer Science*, pages 47–61. Springer, Berlin, 2008.

18 Ken-ichi Kawarabayashi and Bojan Mohar. Graph and map isomorphism and all polyhedral embeddings in linear time. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 471–480. ACM, 2008.

19 Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A Simpler Linear Time Algorithm for Embedding Graphs into an Arbitrary Surface and the Genus of Graphs of Bounded Tree-Width. In *Proceedings of the 49th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 771–780. IEEE Computer Society, 2008.

20 Ken-ichi Kawarabayashi and Bruce A. Reed. Odd cycle packing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010*, pages 695–704, 2010.

21 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear Kernels and Single-Exponential Algorithms Via Protrusion Decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016. `doi:10.1145/2797140`.

22 Dániel Marx. Chordal Deletion is Fixed-Parameter Tractable. *Algorithmica*, 57(4):747–768, 2010. `doi:10.1007/s00453-008-9233-8`.

23 Dániel Marx and Ildikó Schlotter. Obtaining a Planar Graph by Vertex Deletion. *Algorithmica*, 62(3-4):807–822, 2012.

24 Neil Robertson and Paul D. Seymour. Graph Minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(2):92–114, 1986.

25 Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The disjoint paths problem. *J. Comb. Theory Ser. B*, 63(1):65–110, 1995.