The role of planarity in connectivity problems parameterized by treewidth

Julien Baste and Ignasi Sau

AlGCo project-team, CNRS, LIRMM, Montpellier, France.
jbaste@ens-cachan.fr, ignasi.sau@lirmm.fr

Abstract. For some years it was believed that for "connectivity" problems such as Hamiltonian Cycle, algorithms running in time $2^{O(\mathbf{tw})} \cdot n^{O(1)}$ –called *single-exponential*– existed only on planar and other sparse graph classes, where $\mathbf{tw}$ stands for the treewidth of the n-vertex input graph. This was recently disproved by Cygan *et al.* [FOCS 2011] and Bodlaender *et al.* [ICALP 2013], who provided single-exponential algorithms on general graphs for essentially all connectivity problems that were known to be solvable in single-exponential time on sparse graphs. In this article we further investigate the role of planarity in connectivity problems parameterized by treewidth, and convey that several problems can indeed be distinguished according to their behavior on planar graphs. In particular, we show that there exist problems that *cannot* be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ on general graphs but that can be solved in time $2^{O(\mathbf{tw})} \cdot n^{O(1)}$ when restricted to planar graphs, and problems that can be solved in time $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ on general graphs but that *cannot* be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ even when restricted to planar graphs, the negative results holding unless the ETH fails. We feel that our results constitute a first step in a subject that can be much exploited.

Keywords: parameterized complexity, treewidth, connectivity problems, single-exponential algorithms, planar graphs, dynamic programming.

1 Introduction

Motivation and previous work. Treewidth is a fundamental graph parameter that, loosely speaking, measures the resemblance of a graph to a tree. It was introduced by Robertson and Seymour in the early stages of their monumental Graph Minors project [17], but its algorithmic importance originated mainly in Courcelle's theorem [3], stating that any graph problem that can be expressed in CMSO logic can be solved in time $f(\mathbf{tw}) \cdot n$ on graphs with n vertices and treewidth $\mathbf{tw}$. Nevertheless, the function $f(\mathbf{tw})$ given by Courcelle's theorem is unavoidably huge [10], so from an algorithmic point of view it is crucial to identify problems for which $f(\mathbf{tw})$ grows *moderately* fast.

Many problems can be solved in time $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ when the n-vertex input (general) graph comes equipped with a tree-decomposition of width $\mathbf{tw}$. Intuitively, this is the case of problems that can be solved via dynamic programming on a tree-decomposition by enumerating all *partitions* or *packings* of the vertices in the bags of the tree-decomposition, which are $\mathbf{tw}^{O(\mathbf{tw})} = 2^{O(\mathbf{tw} \log \mathbf{tw})}$ many. In this article we only consider this type of problems and, more precisely, we are interested in which of these problems can be solved in time $2^{O(\mathbf{tw})} \cdot n^{O(1)}$; such a running time is called *single-exponential*. This topic has been object of extensive study during the last decade. Let us briefly overview the main results on this line of research.

It is well known that problems that have *locally checkable certificates*[1], like Vertex Cover or Dominating Set, can be solved in single-exponential time on general

[1] That is, certificates consisting of a constant number of bits per vertex that can be checked by a cardinality check and by iteratively looking at the neighborhoods of the input graph.

graphs. Intuitively, for this problems it is enough to enumerate *subsets* of the bags of a tree-decomposition (rather than partitions or packings), which are $2^{O(\mathbf{tw})}$ many. A natural class of problems that do *not* have locally checkable certificates is the class of so-called *connectivity problems*, which contains for example HAMILTONIAN CYCLE, STEINER TREE, or CONNECTED VERTEX COVER. These problems have the property that the solutions should satisfy a *connectivity* requirement (see [1,4,19] for more details), and using classical dynamic programming techniques it seems that for solving such a problem it is necessary to enumerate partitions or packings of the bags of a tree-decomposition.

A series of articles provided single-exponential algorithms for connectivity problems when the input graphs are restricted to be sparse, namely planar [9], of bounded genus [7, 19], or excluding a fixed graph as a minor [8, 20]. The common key idea of these works is to use special types of branch-decompositions (which are objects similar to tree-decompositions) with nice combinatorial properties, which strongly rely on the fact that the input graph is sparse.

Until very recently, it was a common belief that all problems solvable in single-exponential time of general graphs should have locally checkable certificates, specially after Lokshtanov *et al.* [16] proved that one connectivity problem, namely DISJOINT PATHS, cannot be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ on general graphs unless the Exponential Time Hypothesis (ETH) fails[2]. This credence was disproved by Cygan *et al.* [4], who provided single-exponential *randomized* algorithms on general graphs for several connectivity problems, like LONGEST PATH, FEEDBACK VERTEX SET, or CONNECTED VERTEX COVER. More recently, Bodlaender *et al.* [1] presented single-exponential *deterministic* algorithms for basically the same connectivity problems. These two results have been considered a breakthrough, and in particular they imply that essentially all connectivity problems that were known to be solvable in single-exponential time on sparse graph classes [7–9, 19, 20] are also solvable in single-exponential time on general graphs [1, 4].

Our main results. In view of the above discussion, a natural conclusion is that sparsity may not be particularly helpful or relevant for obtaining single-exponential algorithms. However, in this article we convey that sparsity (in particular, planarity) *does* play a role in connectivity problems parameterized by treewidth. To this end, among the problems that can be solved in time $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ on general graphs, we distinguish the following disjoint types:

- **Type 1**: Problems that can be solved in time $2^{O(\mathbf{tw})} \cdot n^{O(1)}$ on general graphs.
- **Type 2:** Problems that *cannot* be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ on general graphs unless the ETH fails, but that can be solved in time $2^{O(\mathbf{tw})} \cdot n^{O(1)}$ when restricted to planar graphs.
- **Type 3:** Problems that *cannot* be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ even when restricted to planar graphs, unless the ETH fails.

Our main contribution is to show that there exist problems of Type 2 and Type 3, thus demonstrating that some connectivity problems can indeed be distinguished according to their behavior on planar graphs. More precisely, we prove the following results:

- It is known that for some problems a single-exponential running time is best possible unless the ETH fails [13]. Nevertheless, such a result requires an ad-hoc proof for each problem. We prove in Section 2 that 3-COLORABILITY, which is a problem of Type 1, cannot be solved in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ unless the ETH fails, even when the input is a planar graph of maximum degree at most 5.

[2] The ETH states that 3-SAT cannot be solved in subexponential time.

- In Section 3 we show that Cycle Packing and some other problems are of Type 2 (the lower bound had already been proved in [4]). Furthermore, we prove that Planar Cycle Packing cannot be solved in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ unless the ETH fails.
- In Section 4 we provide an example of problem of Type 3: Monochromatic Disjoint Paths, which is a variant of the Disjoint Paths problem on a vertex-colored graph with additional restrictions on the allowed colors for each path. To the best of our knowledge, problems of this type had not been identified before.

In order to obtain our results, for the upper bounds we strongly follow the algorithmic techniques based on *Catalan structures* used in [7–9, 19, 20], and for some of the lower bounds we use the framework introduced in [16] and that has been also used in [4]. Due to space limitations, the proofs of the results marked with '[$\star$]' have been moved to the appendix.

Additional results and further research. We feel that our results about the role of planarity in connectivity problems parameterized by treewidth are just a first step in a subject that can be much exploited, and we think that the following avenues are particularly interesting:

- It is known that Disjoint Paths can be solved in time $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ on general graphs [21], and that this bound is asymptotically tight under the ETH [16]. The fact whether Disjoint Paths belongs to Type 2 or Type 3 (or maybe even to some other type in between) remains an important open problem. Towards a possible answer to this question, we prove in Appendix I that Planar Disjoint Paths cannot be solved in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ unless the ETH fails.
- Another fundamental problem is Subgraph Isomorphism, which is known to be solvable in time $2^{O(h)} \cdot n^{O(1)}$ on planar graphs [6] and graphs on surfaces [2], where h is the number of vertices of a pattern graph H to be found in a host graph G on n vertices. We prove in Appendix J that Planar Subgraph Isomorphism cannot be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ unless the ETH fails, but an algorithm running in time $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ (that is, with no dependency on H) is not known to exist.
- Lokshtanov *et al.* [15] have proved that for a number of problems such as Dominating Set or q-Coloring, the best known constant c in algorithms of the form $c^{\mathbf{tw}} \cdot n^{O(1)}$ on general graphs is best possible unless the Strong ETH fails. Is it possible to provide better constants for these problems on planar graphs? The existence of such algorithms would permit to further refine the problems belonging to Type 1.
- Are there NP-hard problems solvable in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$?
- Finally, it would be interesting to obtain similar results for problems parameterized by pathwidth, and to extend our algorithms to more general classes of sparse graphs.

Notation. We use standard graph-theoretic notation, and the reader is referred to [5] for any undefined term. All the graphs we considered are undirected and contain neither loops nor multiple edges. Throughout the paper, when the problem under consideration is clear, we let n denote the number of vertices of the input graph, $\mathbf{tw}$ its treewidth, and $\mathbf{pw}$ its pathwidth. We use the notation $[k]$ for the set of integers $\{1, \ldots, k\}$. In the set $[k] \times [k]$, a *row* is a set $\{i\} \times [k]$ and a *column* is a set $[k] \times \{i\}$ for some $i \in [k]$. If $\mathbf{P}$ is a problem defined on graphs, we denote by Planar $\mathbf{P}$ the restriction of $\mathbf{P}$ to planar input graphs. The definition of tree-, path-, and branch-decomposition, as well as of non-crossing partition and matching, can be found in Appendix A.

2 Tight Problems of Type 1

It is usually believed that NP-hard problems parameterized by **tw** cannot be solved in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ under some reasonable complexity assumption. This has been proved in [13] for problems on *general* graphs such as q-COLORABILITY, INDEPENDENT SET, or VERTEX COVER, assuming the ETH. To the best of our knowledge, such lower bounds are not known for 3-COLORABILITY or CYCLE PACKING when the input graph is restricted to be *planar*. In this section we show that PLANAR 3-COLORABILITY cannot be solved in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ even when the input graph has maximum degree at most 5, and this result will be used to deal with PLANAR CYCLE PACKING in Section 3.

3-COLORABILITY **Input:** An n-vertex graph $G = (V, E)$. **Question:** Is there a coloring $c : V \to \{1, 2, 3\}$ s.t. for all $\{x, y\} \in E$, $c(x) \neq c(y)$?

Theorem 1. [$\star$] PLANAR 3-COLORABILITY *cannot be solved in time* $2^{o(\sqrt{n})} \cdot n^{O(1)}$ *unless the ETH fails, even when the input graph has maximum degree at most 5.*

Corollary 1. PLANAR 3-COLORABILITY *cannot be solved in time* $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ *unless the ETH fails, even if the input graph has maximum degree at most 5.*

Proof: As a planar graph G on n vertices satisfies $\mathbf{tw}(G) = O(\sqrt{n})$ [11], an algorithm in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ for PLANAR 3-COLORABILITY implies that there is an algorithm in time $2^{o(\sqrt{n})} \cdot n^{O(1)}$, which is impossible by Theorem 1 unless the ETH fails. □

3 Problems of Type 2

In this section we prove that the CYCLE PACKING problem is of Type 2. Other problems of Type 2 are discussed in Appendix E.

CYCLE PACKING **Input:** An n-vertex graph $G = (V, E)$ and an integer ℓ_0. **Parameter:** The treewidth **tw** of G. **Question:** Does G contain ℓ_0 pairwise vertex-disjoint cycles?

The proof of the following lemma is a direct application of the techniques introduced in [9], which are based on *Catalan structures*.

Lemma 1. [$\star$] PLANAR CYCLE PACKING *can be solved in time* $2^{O(\mathbf{tw})} \cdot n^{O(1)}$.

Theorem 2. PLANAR CYCLE PACKING *cannot be solved in time* $2^{o(\sqrt{n})} \cdot n^{O(1)}$ *unless the ETH fails. Therefore,* PLANAR CYCLE PACKING *cannot be solved in time* $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ *unless the ETH fails.*

Proof: To prove this theorem, we reduce from PLANAR 3-COLORABILITY where the input graph has maximum degree at most 5. Let $G = (V, E)$ be a planar graph with maximum degree at most 5 with $V = \{v_1, \ldots, v_n\}$. We proceed to construct a planar graph H together with a planar embedding of it, where we will ask for an appropriate number ℓ_0 of vertex-disjoint cycles.

In this proof, we abuse notation and say that we *ask* for x cycles in a gadget to say that the number of cycles we are looking for in the PLANAR CYCLE PACKING problem is increased by x. We will ask for a certain number of cycles in each of the introduced gadgets, which by construction will lead to a set of cycles of maximum cardinality in H.

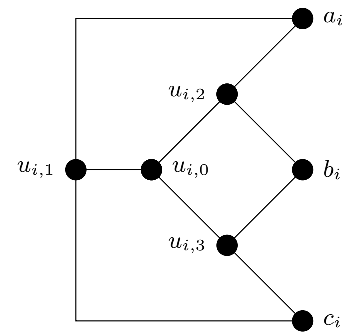
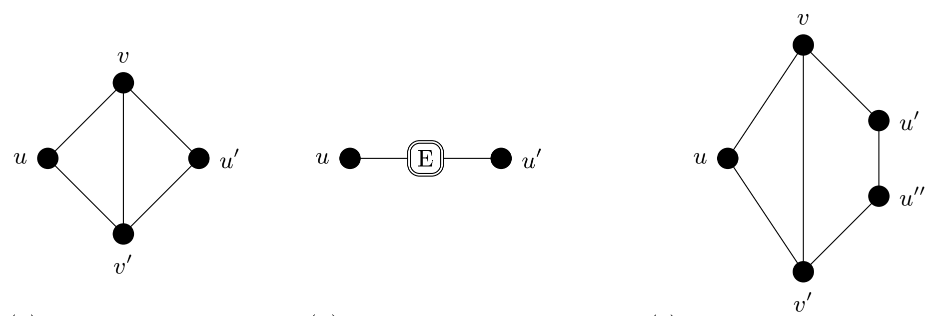


Fig. 1. The SC$_i$-gadget.

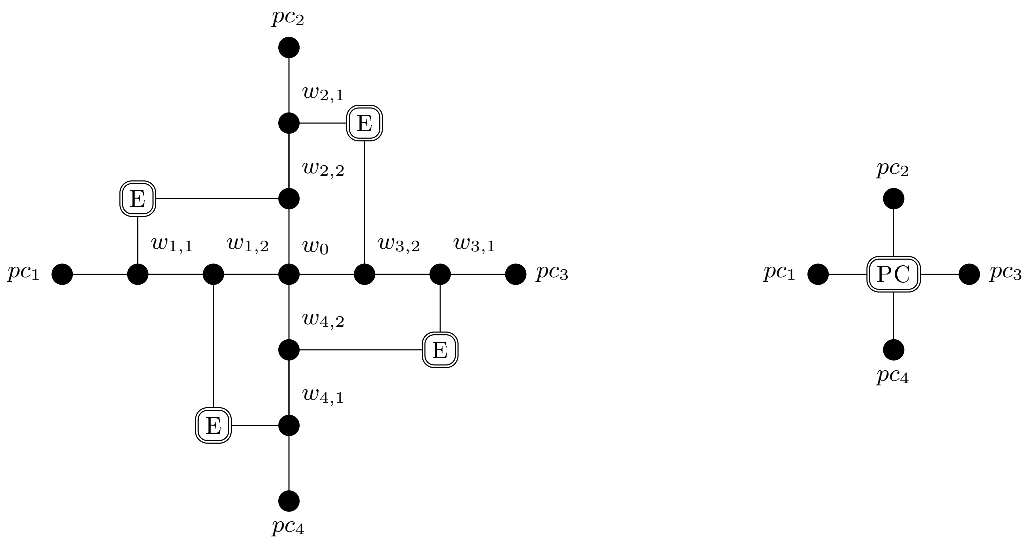


(a) The expel gadget. (b) Its representation. (c) The double-expel gadget.

Fig. 2. The expel gadget and the double-expel gadget.

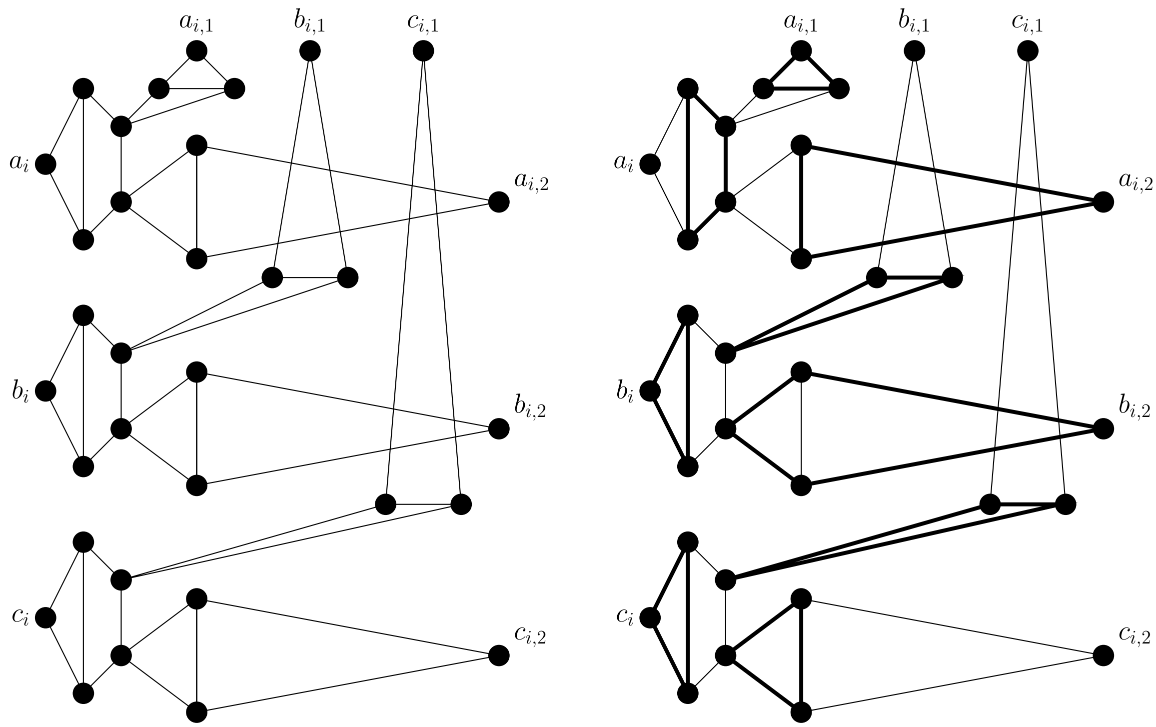
We start by introducing some gadgets. For each $i \in [n]$, corresponding to the vertices $v_1, \ldots, v_n$ of G, we add to H the SC$_i$-gadget depicted in Fig. 1. More precisely, $SC_i = (\{a_i, b_i, c_i, u_{i,0}, u_{i,1}, u_{i,2}, u_{i,3}\}, \{(u_{i,0}, u_{i,1}), (u_{i,0}, u_{i,2}), (u_{i,0}, u_{i,3}), (a_i, u_{i,1}), (a_i, u_{i,2}), (b_i, u_{i,2}), (b_i, u_{i,3}), (c_i, u_{i,1}), (c_i, u_{i,3})\})$. We ask for a cycle inside this gadget. This cycle imposes that at least one of the vertices $\{a_i, b_i, c_i\}$, named a *selected vertex* of the SC$_i$-gadget, is used by the inner cycle and leaves the possibility that the two others are free. The intended meaning of each SC$_i$-gadget is as follows. The three vertices a_i, b_i, and c_i correspond to the three colors in the 3-coloring of G, namely a, b, and c. If for instance a_i is a selected vertex for i, it will imply that vertex v_i can be colored with color a. Therefore, each SC$_i$-gadget defines the available colors for vertex v_i, which we call the *color output* of vertex v_i.

In order to construct a graph H that defines a valid 3-coloring of G, we need to propagate the color output of v_i as many times as the degree of v_i in G. For this, we introduce a gadget called *bifurcate* gadget. Before proceeding to the description of the gadget, let us describe its intended functionality. The objective is, starting with the vertices a_i, b_i, and c_i of the SC$_i$-gadget, to construct a set of triples $\{a_{i,k}, b_{i,k}, c_{i,k}\}$ for $1 \leqslant k \leqslant \deg_G(v_i)$ such that in each triple there will be again at least one selected vertex, defined by the cycles that we will construct in the bifurcate gadgets. Note that in the SC$_i$-gadget the choice of a selected vertex in each triple $\{a_{i,k}, b_{i,k}, c_{i,k}\}$ naturally defines a color output for vertex v_i. The crucial property of the gadget is that the intersection of the color outputs given by all the triples is non-empty if and only if the graph H contains enough vertex-disjoint cycles. In other words, the existence of the appropriate



The gadget. The representation.

Fig. 3. Path-crossing gadget.



(a) The gadget. (b) An example.

Fig. 4. Bifurcate gadget: To keep planarity, there is a path-crossing gadget in each edge intersection.

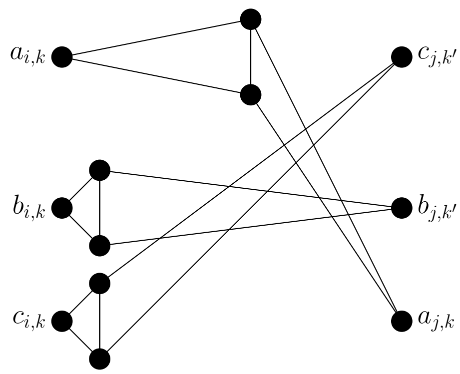


Fig. 5. Edge gadget: To keep planarity, there is a path-crossing gadget in each edge intersection.

number of vertex-disjoint cycles in H will define an available color for each vertex v_i of G.

We now proceed to the construction of the bifurcate gadget. First we need to introduce three other auxiliary gadgets. The first two ones, called *expel* and *double-expel* gadgets, are depicted in Fig. 2. Formally, for two vertices u and u', the expel gadget is defined as $EG_{u,u'} = (\{u, u', v, v'\}, \{(u, v), (u, v'), (u', v), (u', v'), (v, v')\})$, and we ask for a cycle inside each such expel gadget. This gadget ensures that if u is in another cycle, then u' is necessarily used by the internal cycle and vice-versa. Similarly, the double-expel gadget for three vertices u, u', and u'' is defined as $DEG_{u,u',u''} = (\{u, u', u'', v, v'\}, \{(u, v), (u, v'), (u', v), (u'', v'), (u', u''), (v, v')\})$, and we also ask for a cycle inside each such gadget. This gadget ensures that if u is in another cycle, then u' and u'' are necessarily used by the internal cycle and that if u' or u'' are in an external cycle, then u is necessarily used by the internal cycle.

As in our construction the edges of the expel gadgets will cross, we need a gadget that replaces each edge-crossing with a planar subgraph while preserving the existence of the original edges, in the sense that each of the crossing edges gets replaced by a path joining the endvertices of the original edge. This gadget is called *path-crossing* gadget and is depicted in Fig. 3. Formally the path-crossing gadget PCG is such that $\{pc_1, pc_2, pc_3, pc_4, w_0, w_{1,1}, w_{1,2}, w_{2,1}, w_{2,2}, w_{3,1}, w_{3,2}, w_{4,1}, w_{4,2}\} \subseteq V(PCG)$, $E(PCG)$ contains two paths $pc_1, w_{1,1}, w_{1,2}, w_0, w_{3,2}, w_{3,1}, pc_3$ and $pc_2, w_{2,1}, w_{2,2}, w_0, w_{4,2}, w_{4,1}, pc_4$, and we add 4 expel gadgets $EG_{w_{1,1},w_{2,2}}$, $EG_{w_{2,1},w_{3,2}}$, $EG_{w_{3,1},w_{4,2}}$, $EG_{w_{4,1},w_{1,2}}$ to PCG. We ask in this gadget only the 4 cycles asked in the expel gadgets. This gadget ensures that, in order to have enough vertex-disjoint cycles, an external cycle that contains an edge from a path-crossing gadget should go *straight*, i.e., for all $\alpha \in [4]$, if the cycle arrives at a vertex pc_α it should leave by $pc_{(\alpha+1 \pmod 4))+1}$. If a cycle does not respect this property, we say that the cycle *turns* inside the path-crossing gadget. That is, the gadget preserves the existence of the original crossing edges whenever there are no cycles that turn inside it. Note that the two paths corresponding to the two original crossing edges cannot be used simultaneously by a set of cycles in the planar graph H. We can now define the bifurcate gadget, which is depicted in Fig. 4(a), and where each of the 12 edge-crossings should be replaced by a path-crossing gadget. Note that each bifurcate gadgets contains 6 expel and 3 double-expel gadgets. We ask in this gadget the 48 cycles of the path-crossing gadgets, the 3 cycles of the double expel gadgets, and the 6 cycles of the expel gadgets. Note that, indeed, given a triple $\{a_i, b_i, c_i\}$ defining a color output for a vertex v_i, the cycles asked in the bifurcate gadget define two triples $\{a_{i,1}, b_{i,1}, c_{i,1}\}$ and $\{a_{i,2}, b_{i,2}, c_{i,2}\}$, which in turn define two color outputs compatible with the one defined by $\{a_i, b_i, c_i\}$, in the sense that there is a common available color for v_i. For example, in Fig. 4(b) vertex a_i is the only selected vertex of $\{a_i, b_i, c_i\}$ (given by the corresponding SC_i-gadget which is not shown in the figure for the sake of visibility), and the bold cycles define the selected vertices for the triples $\{a_{i,1}, b_{i,1}, c_{i,1}\}$ and $\{a_{i,2}, b_{i,2}, c_{i,2}\}$. Note that color a is simultaneously available for the three triples. We would like to stress that there are other choices of a maximum-cardinality set of cycles in the bifurcate gadget of Fig. 4(b), but all of them yield color a available. For each vertex v_i, we need as many triples $\{a_{i,k}, b_{i,k}, c_{i,k}\}$ as $\deg_G(v_i)$. For that, we concatenate the bifurcate gadgets $deg_G(v_i) - 1$ times in the following way. Inductively, we consider the triple $\{a_{i,2}, b_{i,2}, c_{i,2}\}$ of Fig. 4(a) as the original triple $\{a_i, b_i, c_i\}$, and plug another bifurcate gadget starting from this triple.

With the gadgets defined so far, we have a representation of the colored vertices of G in H. We now proceed to capture the edges of G in H. For this, we introduce for each $\{v_i, v_j\} \in E$, $i, j \in [n]$, an *edge* gadget depicted in Fig. 5, where all the 12 edge-crossings should be replaced by a path-crossing gadget. We ask in this gadget 51 new cycles (3 for the expel gadgets and 48 for the path-crossing gadgets). We plug one side of this gadget to a triple $\{a_{i,k}, b_{i,k}, c_{i,k}\}$ defining a color output of v_i and the other side to a triple $\{a_{j,k'}, b_{j,k'}, c_{j,k'}\}$ defining a color output of v_j. The edge gadget ensures that the intersection of the two color outputs is empty. This completes the construction of H, which is clearly a planar graph, and we set ℓ_0 to be the sum of the number of cycles asked in each of the introduced gadgets.

Claim 1 [$\star$] *In any solution of* CYCLE PACKING *in H, each expel gadget, double-expel gadget, and SC_i-gadget contains a cycle, and each cycle is contained inside such a gadget.*

If we are given a solution of PLANAR CYCLE PACKING in H, then for each $i \in [n]$, by Claim 1 the selection of a cycle in the SC_i-gadget selects a color for v_i, that can be any color that belongs simultaneously to all color outputs of v_i, and the edge gadgets ensure that two adjacent vertices are in two different color classes. So in this way we obtain a solution of PLANAR 3-COLORABILITY in G.

Conversely, given a solution of PLANAR 3-COLORABILITY in G, we construct a solution of PLANAR CYCLE PACKING in H as follows. For each $i \in [n]$ we choose in the SC_i-gadget the cycle of length 4 that contains $u_{i,0}$ and the vertex in $\{a_i, b_i, c_i\}$ that corresponds to the color of v_i. We also choose in the bifurcates gadgets the cycles selecting vertices in $\{a_{i,1}, b_{i,1}, c_{i,1}, a_{i,2}, b_{i,2}, c_{i,2}\}$ that lead to two identical color outputs coinciding with the color output of $\{a_i, b_i, c_i\}$. This choice has the property that the color output of $\{a_i, b_i, v_i\}$ is a subset of the color output of $\{a_{i,1}, b_{i,1}, c_{i,1}\}$ and the color output of $\{a_{i,2}, b_{i,2}, c_{i,2}\}$, and leaves as many free vertices as possible for other cycles in other gadgets. Inside the edge gadget representing $\{v_i, v_j\} \in E$, we select the three cycles that are allowed by the free vertices. We complete our cycle selection by selecting a cycle in each expel gadget contained in a path-crossing gadget. By Claim 1, this choice leads to a solution of PLANAR CYCLE PACKING in H.

As the degree of each vertex in G is bounded by 5, the number of gadgets we introduce for each $v_i \in V(G)$ to construct H is also bounded by a constant, so the total number of vertices of H is linear in the number of vertices of G. Therefore if we could solve PLANAR CYCLE PACKING in time $2^{o(\sqrt{n})} \cdot n^{O(1)}$ then we could also solve PLANAR 3-COLORING in time $2^{o(\sqrt{n})} \cdot n^{O(1)}$, which is impossible by Theorem 1 unless the ETH fails. □

4 Problems of Type 3

In this section we prove that the MONOCHROMATIC DISJOINT PATHS problem is of Type 3. We first need to introduce some definitions. Let $G = (V, E)$ be a graph, let k

be an integer, and let $c : V \to \{0, \dots, k\}$ be a color function. Two colors c_1 and c_2 in $\{0, \dots, k\}$ are *compatible*, and we denote it by $c_1 \equiv c_2$, if $c_1 = 0$, $c_2 = 0$, or $c_1 = c_2$. A path $P = x_1 \dots x_m$ in G is *monochromatic* if for all $i, j \in [m]$, $i \neq j$, $c(x_i)$ and $c(x_j)$ are two compatible colors. We let $c(P) = \max_{i \in [m]}(c(x_i))$. We say that P is *colored* x if $x = c(P)$. Two monochromatic paths P and P' are *color-compatible* if $c(P) \equiv c(P')$.

> MONOCHROMATIC DISJOINT PATHS
> **Input:** A graph $G = (V, E)$ of treewidth $\mathbf{tw}$, a color function $\gamma : V \to \{0, \dots, \mathbf{tw}\}$, an integer m, and a set $\mathcal{N} = \{\mathcal{N}_i = \{s_i, t_i\} | i \in [m], s_i, t_i \in V\}$.
> **Parameter:** The treewidth $\mathbf{tw}$ of G.
> **Question:** Does G contain m pairwise vertex-disjoint monochromatic paths from s_i to t_i, for $i \in [m]$?

The proof of the following lemma is inspired from the algorithm given in [21] for the DISJOINT PATHS problem on general graphs.

Lemma 2. [$\star$] MONOCHROMATIC DISJOINT PATHS *can be solved in time* $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$.

We need to define the $k \times k$-HITTING SET problem, first introduced in [16].

> $k \times k$-HITTING SET
> **Input:** A family of sets $S_1, S_2, \dots, S_m \subseteq [k] \times [k]$, such that each set contains at most one element from each row of $[k] \times [k]$.
> **Parameter:** k.
> **Question:** Is there a set S containing exactly one element from each row such that $S \cap S_i \neq \emptyset$ for any $1 \leqslant i \leqslant m$?

Theorem 3 (Lokshtanov *et al.* [16]). $k \times k$-HITTING SET *cannot be solved in time* $2^{o(k \log k)} \cdot m^{O(1)}$ *unless the ETH fails.*

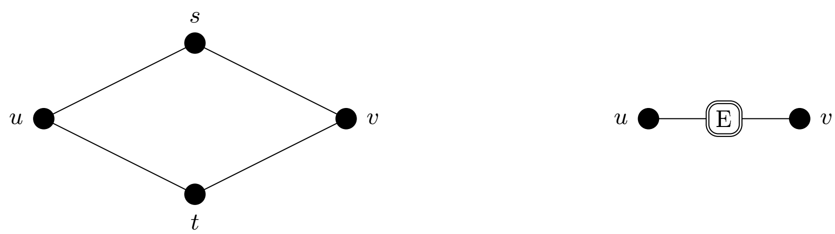
We state the following theorem in terms of the pathwidth of the input graph, and as any graph G satisfies $\mathbf{tw}(G) \leqslant \mathbf{pw}(G)$, it implies the same lower bound in the treewidth.

Theorem 4. PLANAR MONOCHROMATIC DISJOINT PATHS *cannot be solved in time* $2^{o(\mathbf{pw} \log \mathbf{pw})} \cdot n^{O(1)}$ *unless the ETH fails.*

Proof: We reduce from $k \times k$-HITTING SET. Let k be an integer and $S_1, S_2, \dots, S_m \subseteq [k] \times [k]$ such that each set contains at most one element from each row of $[k] \times [k]$. We will first present an overview of the reduction with all the involved gadgets, and then we will provide a formal definition of the constructed planar graph G.

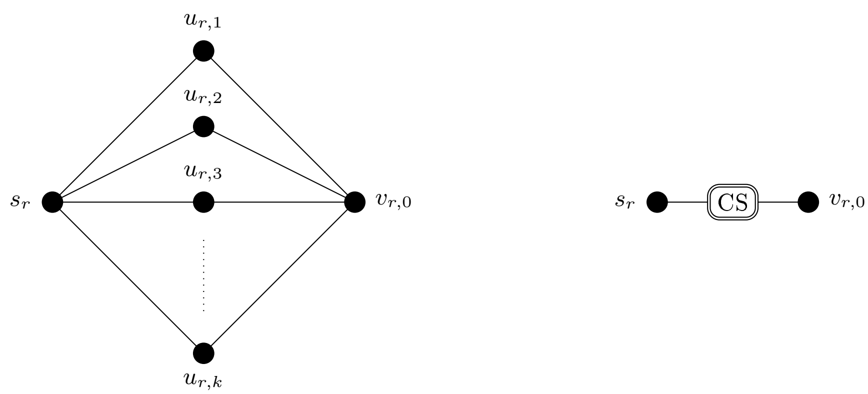
We construct a gadget for each row $\{r\} \times [k]$, $r \in [k]$, which selects the unique pair p of S in this row. First, for each $r \in [k]$, we introduce two new vertices s_r and t_r, a request $\{s_r, t_r\}$, $m + 1$ vertices $v_{r,i}$, $i \in \{0, \dots, m\}$, and $m + 2$ edges $\{e_{r,0} = (s_r, v_{r,0})\} \cup \{e_{r,i} = (v_{r,i-1}, v_{r,i}) | i \in [m]\} \cup \{e_{r,m+1} = (v_{r,m}, t_r)\}$. That is, we have a path with $m + 2$ edges between s_r and t_r.

Each edge of these paths, except the last one, will be replaced with an appropriate gadget. Namely, for each $r \in [k]$, we replace the edge $e_{r,0}$ with the gadget depicted in Fig. 7, which we call *color-selection* gadget. In this figure, vertex $u_{r,i}$ is colored i. The color used by the path from s_r to t_r in the color-selection gadget will define the pair of the solution of S in the row $\{r\} \times [k]$.



The gadget. The representation.

Fig. 6. Expel gadget.



The gadget. The representation.

Fig. 7. Color-selection gadget, where $u_{r,i}$ is colored c_i for each $i \in [k]$.

Now that we have described the gadgets that allow to define S, we need to ensure that $S \cap S_i \neq \emptyset$ for any $i \in [m]$. For this, we need the gadget depicted in Fig. 6, which we call *expel* gadget. Each time we introduce this gadget, we add to $\mathcal{N}$ the request $\{s, t\}$. This new requested path uses either vertex u or vertex v, so only one of these vertices can be used by other paths. For each $i \in [m]$, we replace all the edges $\{e_{r,i} | r \in [k]\}$ with the gadget depicted in Fig. 8, which we call *set* gadget. In this figure, $a_{r,i}$ is such that if $(\{r\} \times [k]) \cap S_i = \{\{r, c_{r,i}\}\}$ then $a_{r,i}$ is colored $c_{r,i}$, and if $(\{r\} \times [k]) \cap S_i = \emptyset$ then vertex $a_{r,i}$ is removed from the gadget.

This completes the construction of the graph G, which is illustrated in Fig. 13 in Appendix G. The formal description of G can be found in Appendix H. Note that G is indeed planar. The color function γ of G is defined such that for each $r \in [k]$ and $c \in [k]$, $\gamma(u_{r,c}) = c$, and for each $i \in [m]$ and $(r, c) \in S_i$, $\gamma(a_{r,i}) = c$. For any other vertex $v \in V(G)$, we set $\gamma(v) = 0$. Finally, the input of Planar Monochromatic Disjoint Paths is the planar graph G, the color function γ, and the $k + (k-1) \cdot m$ requests $\mathcal{N} = \{\{s_r, t_r\} | r \in [k]\} \cup \{\{s_{r,i}, t_{r,i}\} | r \in [k-1], i \in [m]\}$, the second set of requests corresponding to the ones introduced by the expel gadgets.

Note that because of the expel gadgets, the request $\{s_r, t_r\}$ imposes a path between $v_{r,i-1}$ and $v_{r,i}$ for each $r \in [k]$. Note also that because of the expel gadgets, at least one of the paths between $v_{r,i-1}$ and $v_{r,i}$ should use an $a_{r,i}$ vertex, as otherwise at least two paths would intersect. Conversely, if one path uses a vertex $a_{r,i}$, then we can find all the desired paths in the corresponding set gadgets by using the vertices $w_{r,i,b}$.

Given a solution of Planar Monochromatic Disjoint Paths in G, we can construct a solution of $k \times k$-Hitting Set by letting $S = \{(r, c) | r \in [k]$ such that the path

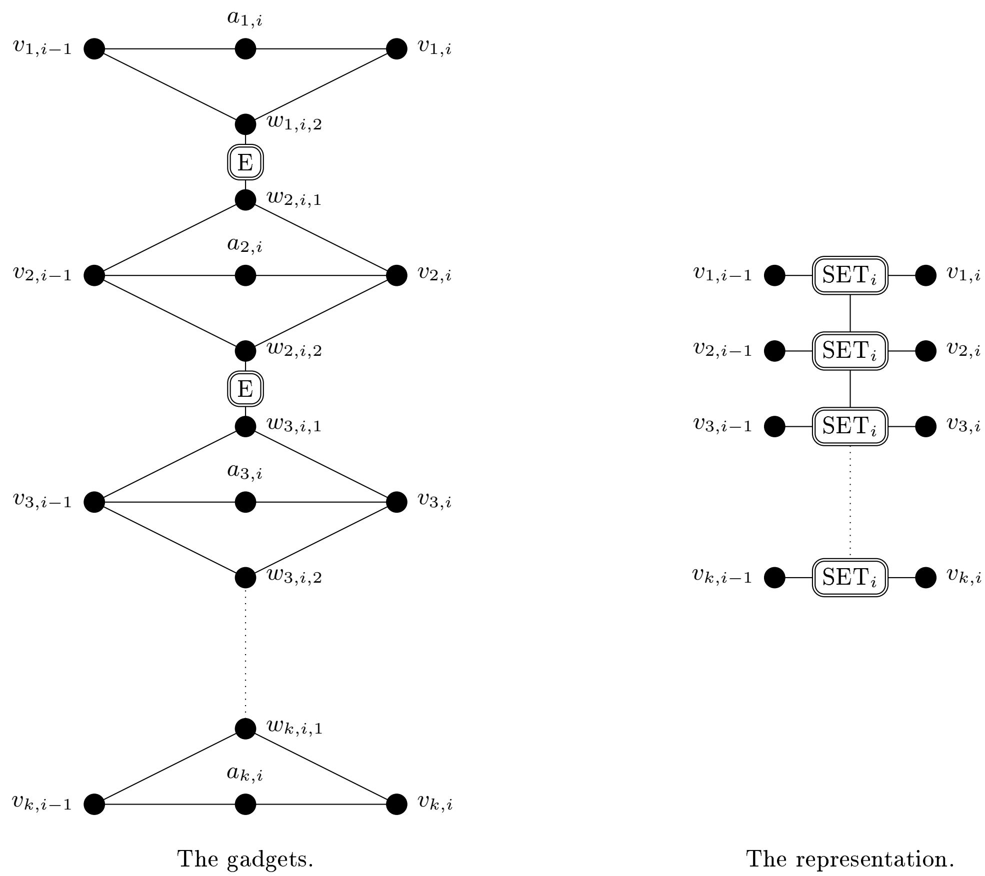


Fig. 8. Set gadgets.

from s_r to t_r is colored with color $c\}$. We have that S contains exactly one element of each row, so we just have to check if $S \cap S_i \neq \emptyset$ for each $i \in [m]$. Because of the property of the set gadgets mentioned above, for each $i \in [m]$, the set gadget labeled i ensures that $S \cap S_i \neq \emptyset$.

Conversely, given a solution S of $k \times k$-Hitting Set, for each $\{r, c\} \in S$ we color the path from s_r to t_r with color c. We assign an arbitrary coloring to the other paths. For each $i \in [m]$, we take $\{r, c\} \in S \cap S_i$ and in the set gadget labeled i, we impose that the path from $v_{r,i-1}$ to $v_{r,i}$ uses vertex $a_{r,i}$. By using the vertices $w_{r,i,b}$ for the other paths, we find the desired $k + (k-1) \cdot m$ monochromatic paths.

Let us now argue about the pathwidth of G. We define for each $r, c \in [k]$ the bag $B_{0,r,c} = \{s_{r'} | r' \in [k]\} \cup \{v_{r',0} | r' \in [k]\} \cup \{u_{r,c}\}$, for each $i \in [m]$, the bag $B_i = \{v_{r,i-1} | r \in [k]\} \cup \{v_{r,i} | r \in [k]\} \cup \{a_{r,i} \in V(G) | r \in [k]\} \cup \{w_{r,i,b} \in V(G) | r \in [k], b \in [2]\} \cup \{s_{r,i} | r \in [m-1]\} \cup \{t_{r,i} | r \in [m-1]\}$, and the bag $B_{m+1} = \{v_{r,m} | r \in [k]\} \cup \{t_r | r \in [k]\}$. We note that the size of each bag is at most $2 \cdot (k-1) + 5 \cdot k - 2 = O(k)$. A path decomposition of G consists of all bags $B_{0,r,c}$, $r, c \in [k]$ and B_i, $i \in [m+1]$ and edges $\{B_i, B_{i+1}\}$ for each $i \in [m]$, $\{B_{0,r,c}, B_{0,r,c+1}\}$ for $r \in [k]$, $c \in [k-1]$, $\{B_{0,r,k}, B_{0,r+1,1}\}$ for $r \in [k]$, and $\{B_{0,k,k}, B_1\}$. Therefore, as we have that $\mathbf{pw}(G) = O(k)$, if one could solve Planar Monochromatic Disjoint Paths in time $2^{o(\mathbf{pw} \log \mathbf{pw})} \cdot n^{O(1)}$, then one could also solve $k \times k$-Hitting Set in time $2^{o(k \log k)} \cdot m^{O(1)}$, which is impossible by Theorem 3 unless the ETH fails. □

References

1. H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Solving weighted and counting variants of connectivity problems parameterized by treewidth deterministically in single exponential time. *CoRR*, abs/1211.1505, 2012, to appear in *Proc. of ICALP'13*.
2. P. Bonsma. Surface split decompositions and subgraph isomorphism in graphs on surfaces. In *Proc. of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 14 of *LIPIcs*, pages 531–542, 2012.
3. B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
4. M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time. In *Proc. of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 150–159. IEEE Computer Society, 2011.
5. R. Diestel. *Graph Theory.* Springer-Verlag, 3rd edition, 2005.
6. F. Dorn. Planar subgraph isomorphism revisited. In *Proc. of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 5 of *LIPIcs*, pages 263–274, 2010.
7. F. Dorn, F. V. Fomin, and D. M. Thilikos. Fast Subexponential Algorithm for Non-local Problems on Graphs of Bounded Genus. In *Proc. of the 10th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 4059 of *LNCS*, pages 172–183, 2006.
8. F. Dorn, F. V. Fomin, and D. M. Thilikos. Catalan Structures and Dynamic Programming in H-minor-free graphs. *Journal of Computer and System Sciences*, 78(5):1606–1622, 2012.
9. F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient Exact Algorithms on Planar Graphs: Exploiting Sphere Cut Decompositions. *Algorithmica*, 58(3):790–810, 2010.
10. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. Springer, 2006.
11. F. V. Fomin and D. M. Thilikos. New upper bounds on the decomposability of planar graphs. *Journal of Graph Theory*, 51(1):53–81, 2006.
12. M. Garey, D. Johnson, and R. E. Tarjan. The Planar Hamiltonian Circuit Problem is NP-Complete. In *SIAM J Comput.*, pages 704–714, 1976.
13. R. Impagliazzo, R. Paturi, and F. Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
14. G. Kreweras. Sur les partitions non croisees d'un cycle. *Discrete Mathematics*, 1(4):333 – 350, 1972.
15. D. Lokshtanov, D. Marx, and S. Saurabh. Known Algorithms on Graphs of Bounded Treewidth are Probably Optimal. In *Proc. of the 22nd annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 777–789, 2011.
16. D. Lokshtanov, D. Marx, and S. Saurabh. Slightly Superexponential Parameterized Problems. In *Proc. of the 22nd annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 760–776, 2011.
17. N. Robertson and P. D. Seymour. Graph Minors. II. Algorithmic Aspects of Tree-Width. *Journal of Algorithms*, 7(3):309–322, 1986.
18. N. Robertson and P. D. Seymour. Graph Minors. X. Obstructions to Tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.
19. J. Rué, I. Sau, and D. M. Thilikos. Dynamic programming for graphs on surfaces. *CoRR*, abs/1104.2486, 2011, to appear in *ACM Transactions on Algorithms (TALG)*. Short version in the *Proc. of ICALP'10*.
20. J. Rué, I. Sau, and D. M. Thilikos. Dynamic Programming for H-minor-free Graphs. In *Proc. of the 18th Annual International Conference on Computing and Combinatorics (COCOON)*, volume 7434 of *LNCS*, pages 86–97, 2012.
21. P. Scheffler. A practical linear time algorithm for disjoint paths in graphs with bounded tree-width. Fachbereich 3 Mathematik, Tech. Report 396/1994, FU Berlin, 1994.
22. P. D. Seymour and R. Thomas. Call Routing and the Ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

A Preliminaries

Graphs. We denote by $V(G)$ the set of vertices of a graph G and by $E(G)$ its set of edges. A *subgraph* $H = (V_H, E_H)$ of a graph $G = (V, E)$ is a graph such that $V_H \subseteq V$ and $E_H \subseteq E \cap (V_H \times V_H)$. The *degree* of a vertex v in a graph G, denoted by $\deg_G(v)$, is the number of edges of G containing v. A *grid* $m * k$ is a graph $Gr_{m,k} = (\{a_{i,j} | i \in [m], j \in [k]\}, \{(a_{i,j}, a_{i+1,j}) | i \in [m-1], j \in [k]\} \cup \{(a_{i,j}, a_{i,j+1}) | i \in [m], j \in [k-1]\})$. When $m = k$ we just speak about a *grid of size* k. We say that there is a *path* $s \dots t$ in a graph G if there exist $m \in \mathbb{N}$ and $x_0, \dots, x_m$ in $V(G)$ such that $x_0 = s$, $x_m = t$, and for all $i \in [m]$, $(x_{i-1}, x_i) \in E(G)$.

Treewidth and pathwidth. A *tree-decomposition* of width w of a graph $G = (V, E)$ is a pair (T, σ), where T is a tree and $\sigma = \{B_t | B_t \subseteq V, t \in V(T)\}$ such that:

- $\bigcup_{t \in V(T)} B_t = V$;
- For every edge $\{u, v\} \in E$ there is a $t \in V(T)$ such that $\{u, v\} \subseteq B_t$;
- $B_i \cap B_k \subseteq B_j$ for all $\{i, j, k\} \subseteq V(T)$ such that j lies on the path $i \dots k$ in T;
- $\max_{i \in V(T)} |B_t| = w + 1$.

The sets B_t are called *bags*. The *treewidth* of G, denoted by $\mathbf{tw}(G)$, is the smallest integer w such that there is a tree-decomposition of G of width w. An *optimal tree-decomposition* is a tree-decomposition of width $\mathbf{tw}(G)$. A *path-decomposition* of a graph $G = (V, E)$ is a tree-decomposition (T, σ) such that T is a path. The *pathwidth* of G, denoted by $\mathbf{pw}(G)$, is the smallest integer w such that there is a path-decomposition of G of width w. Clearly, for any graph G, we have $\mathbf{tw}(G) \leqslant \mathbf{pw}(G)$.

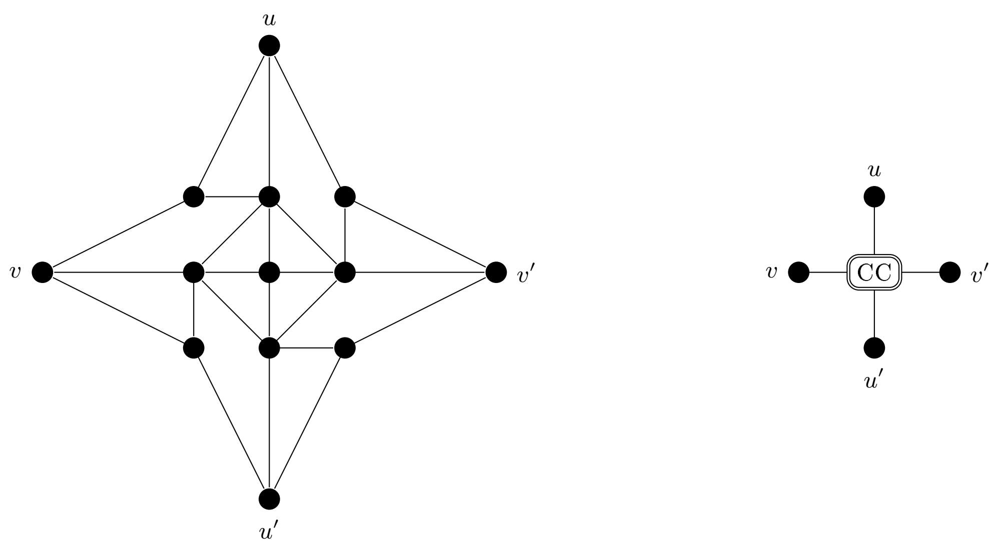
Branchwidth. A *branch-decomposition* (T, σ) of a graph $G = (V, E)$ consists of an unrooted ternary tree T and a bijection $\sigma : L \to E$ from the set L of leaves of T to the edge set of G. We define for every edge e of T the *middle set* $\mathbf{mid}(e) \subseteq V(G)$ as follows: Let T_1 and T_2 be the two connected components of $T \backslash \{e\}$. Then let G_i be the graph induced by the edge set $\{\sigma(f) : f \in L \cap V(T_i)\}$ for $i \in \{1, 2\}$. The *middle set* of e is the intersection of the vertex sets of G_1 and G_2, i.e., $\mathbf{mid}(e) := V(G_1) \cap V(G_2)$. When we consider T as rooted, we let G_e be the graph G_i such that T_i does not contain the root of T. The *width* of (T, σ) is the maximum order of the middle sets over all edges of T, i.e., $w(T, \sigma) := \max\{|\mathbf{mid}(e)| | e \in T\}$. The *branchwidth* of G, denoted by $\mathbf{bw}(G)$, is the minimum width over all branch decompositions of G. An *optimal branch decomposition* of G is a branch decomposition (T, σ) of width $\mathbf{bw}(G)$. By [18], the branchwidth of a graph G with at least 3 edges is related to its treewidth by $\mathbf{bw}(G) - 1 \leqslant \mathbf{tw}(G) \leqslant \lfloor \frac{3}{2} \mathbf{bw}(G) \rfloor - 1$.

Planar graphs. Let Σ be the sphere $\{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1\}$. By a *$\Sigma$-plane* graph G we mean a planar graph G with its vertex set $V(G)$, edge set $E(G)$, and face set $F(G)$ drawn without edge crossings in Σ. An O-arc is a subset of Σ homeomorphic to a circle. An O-arc in Σ is called a *noose* of a Σ-plane graph G if it meets G only in vertices and intersects with every face at most once. Each noose O bounds two open discs Δ_1, Δ_2 in Σ, i.e., $\Delta_1 \cap \Delta_2 = \emptyset$ and $\Delta_1 \cup \Delta_2 \cup O = \Sigma$.

For a Σ-plane graph G, we define a *sphere cut decomposition* (T, σ, π) of G, or *sc-decomposition* for short, as a branch-decomposition such that for every edge e of T there exists a noose O_e bounding the two open discs Δ_1 and Δ_2 such that $G_i \subseteq \Delta_i \cup O_e$, $1 \leqslant i \leqslant 2$. Thus O_e meets G only in $\mathbf{mid}(e)$ and its length is $|\mathbf{mid}(e)|$. It is known that any planar graph G has a sc-decomposition of width $\mathbf{bw}(G)$ that can be computed in polynomial time [9, 22].

The gadget. The representation.

Fig. 9. Color gadget.



The gadget. The representation.

Fig. 10. Cross-color gadget.

Non-crossing partitions and matchings. A *partition* P of a set S is a set of subsets of S such that $\bigcup_{s \in P} s = S$ and for all distinct $s_1, s_2 \in P$, $s_1 \cap s_2 = \emptyset$. A partition P is called *non-crossing partition* if for each $s_1, s_2 \in P$, for each $a, b \in s_1$ and $c, d \in s_2$ with $a < b$ and $c < d$ then one of the following situations occurs: $a < b < c < d$, $a < c < d < b$, $c < d < a < b$, or $c < a < b < d$. Kreweras showed in [14] that the number of non-crossing partitions on $[k]$ for $k \in \mathbb{N}$ is at most 4^k.

A *matching* M is a set of pairs of elements of a set, which we also call edges, such that for each $e, e' \in M$, $e \neq e'$, $e \cap e' = \emptyset$. For a matching M in a graph G, we denote by $V[M]$ the set of all vertices that belong to an edge of M. We say that two matchings M and M' are *disjoint* if $V[M] \cap V[M'] = \emptyset$. A matching M on $V = \{v_1, \ldots, v_n\}$, for some $n \in \mathbb{N}^*$, is called *non-crossing matching* if for each $\{v_a, v_b\}, \{v_c, v_d\} \in M$, with $a < b$ and $c < d$, then one of the following situations occurs: $a < b < c < d$, $a < c < d < b$, $c < d < a < b$, or $c < a < b < d$. Kreweras showed in [14] that the number of non-crossing matchings on $[k]$ for $k \in \mathbb{N}$ is at most 2^k.

B Proof of Theorem 1

We start with defining some planar gadgets. The first one is depicted in Fig. 9 and called *color gadget*, C-*gadget* for short. This gadget ensures that two vertices u and u' are in the

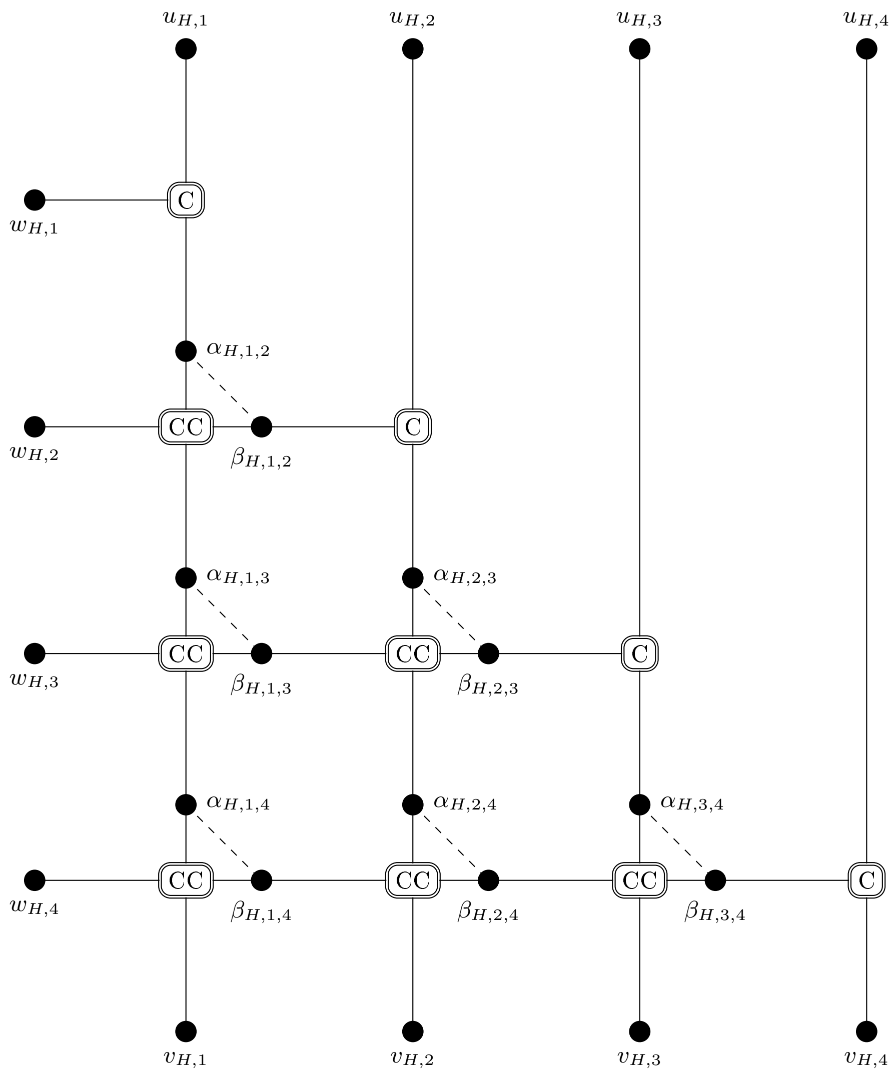
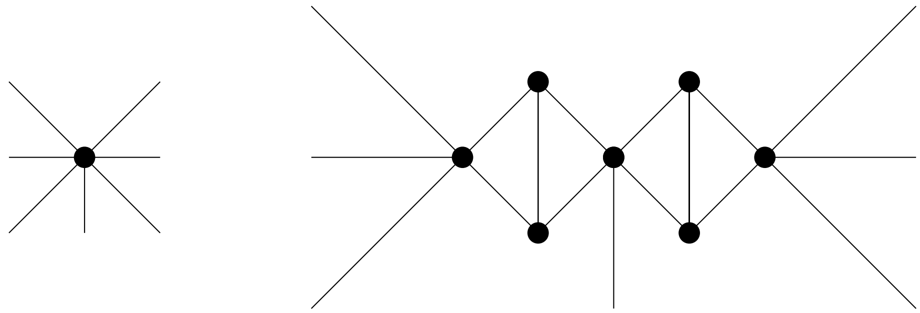


Fig. 11. An example of graph H for $n = 4$.

same color class. Note that we can extend the C-gadget for three vertices u, u', and u'' and ensure the three vertices to be in the same color class by fixing a C-gadget between u and u' and another C-gadget between u' and u''. The second gadget is depicted in Fig. 10 and called *cross-color* gadget, CC-*gadget* for short. In this gadget, originally introduced in [12], one can check that if u, v, u', and v' are in the same face and oriented in this order around the face, that u and u' are in the same color class and v and v' are in the same color class.

We reduce from 3-Colorability. Let $G = (V, E)$ be an input general graph with $n = |V|$ and $V = \{v_1, \ldots, v_n\}$, and we define the planar graph H, illustrated in Fig. 11 for $n = 4$, as follows:

- For each $i \in [n]$, $u_{H,i}, v_{H,i}, w_{H,i} \in V(H)$;
- For each $i, j \in [n]$, $i < j$, $\alpha_{H,i,j} \in V(H)$ and $\beta_{H,i,j} \in V(H)$;
- For each $i \in \{1, \ldots, n-1\}$, there is a C-gadget between $u_{H,i}$ and $\alpha_{H,i-1,i}$;
- For each $i \in \{2, \ldots, n\}$, there is a C-gadget between $u_{H,i}$ and $\beta_{H,i-1,i}$;



The original vertex. Simulation of the same vertex with maximum degree 5.

Fig. 12. Reducing the maximum degree from 7 to 5.

- There is a C-gadget between $u_{H,n}$ and $w_{H,n}$;
- There is a C-gadget between $u_{H,1}$ and $v_{H,1}$;
- For each $i, j \in \{2, \ldots, n-1\}$, $i < j$ there is a CC-gadget between $\alpha_{H,i,j}$, $\beta_{H,i,j}$, $\alpha_{H,i,j+1}$, and $\beta_{H,i-1,j}$;
- For each $i \in \{2, \ldots, n-1\}$, $i < j$ there is a CC-gadget between $\alpha_{H,i,n}$, $\beta_{H,i,n}$, $w_{H,i}$, and $\beta_{H,i-1,n}$;
- For each $j \in \{2, \ldots, n-1\}$, $i < j$ there is a CC-gadget between $\alpha_{H,1,j}$, $\beta_{H,1,j}$, $\alpha_{H,1,j+1}$, and $v_{H,j}$;
- There is a CC-gadget between $\alpha_{H,1,n}$, $\beta_{H,1,n}$, $w_{H,1}$, and $v_{H,n}$;
- For each $i, j \in [n]$, $i < j$, if $(v_i, v_j) \in E$, then $(\alpha_{H,i,j}, \beta_{H,i,j}) \in E(H)$.

As the C-gadget and the CC-gadget are planar, H is indeed planar (see Fig. 11).

Because of the properties on the C-gadget and the CC-gadget, for each i in $[n]$, $u_{H,i}$, $v_{H,i}$, and $w_{H,i}$ are in the same color class. Because of the edges $(\alpha_{H,i,j}, \beta_{H,i,j})$, if there is an edge between v_i and v_j in G, then $u_{H,i}$ and $u_{H,j}$ should receive different colors. If we have a 3-coloring of G, then by coloring $u_{H,i}$ with the color of v_i for each $i \in [n]$, we find a 3-coloring of H. Conversely, if we have a coloring of H, for each $i \in [n]$ we color each vertex v_i of V with the color of $u_{H,i}$.

Let us now argue about the maximum degree of the graph H. With the previous construction described so far, H has maximum degree 7. In order to restrict it to 5, we replace each vertex of degree 6 or 7 with the two color gadgets shown in Fig. 12.

Let us finally argue about the number of vertices of H. Note that H can be seen as a spanning subgraph of a grid of size n, where each vertex either has been replaced by a C-gadget or a CC-gadget, or it has been removed. As these two gadgets have at most 13 vertices, and in the worst case, all these new vertices have degree 7 and we need to replace them with two color gadgets, that have 7 vertices each, we have that $|V(H)| \leqslant 65 \cdot n^2$. As 3-Colorability cannot be solved in time $2^{o(n)} \cdot n^{O(1)}$ unless the ETH fails [13], the theorem follows.

C Proof of Lemma 1

We prove the lemma for $\mathbf{bw}$, but as $\mathbf{tw} \leqslant \lfloor \frac{3}{2}\mathbf{bw} \rfloor - 1$, it will imply the same asymptotic upper bound for $\mathbf{tw}$. Let G be a graph, $X \subseteq V(G)$, and M a matching on $V(G) \backslash X$. Intuitively, M represents the endpoints of the paths we are building and X is the set of vertices that are already inside a path but they are not an endpoint of any path. We define $G[(X, M, \ell)] = (V[M], M)$. We say that $G[(X_1, M_1, \ell_1), (X_2, M_2, \ell_2)]$ is *defined* if

$X_1 \cap (X_2 \cup V[M_2]) = X_2 \cap (X_1 \cup V[M_1]) = \emptyset$ and we define $G[(X_1, M_1, \ell_1), (X_2, M_2, \ell_2)] = G[(X_1, M_1, \ell_1)] \cup G[(X_2, M_2, \ell_2)]$. Otherwise, we say that $G[(X_1, M_1, \ell_1), (X_2, M_2, \ell_2)]$ is *undefined*. We say that $cp(G, X, M) \geqslant \ell$ if G contains paths joining each pair of vertices given by M and ℓ cycles, all pairwise vertex-disjoint.

We now consider $G = (V, E)$ to be our Σ-plane input graph and ℓ_0 our integer. Let (T, μ, π) be a sc-decomposition of G of width $\mathbf{bw}$. As in [9], we root T by arbitrarily choosing an edge e and we subdivide it by inserting a new node s. Let e' and e'' be the new edges and set $\mathbf{mid}(e') = \mathbf{mid}(e'') = \mathbf{mid}(e)$. We create a new node root r, we connect it to s by an edge e_r, and set $\mathbf{mid}(e_r) = \emptyset$. The root e_r is not considered as a leaf.

Let $e \in E(T)$ and $\mathcal{R}_e = \{(X, M, \ell) | X \subseteq \mathbf{mid}(e)$, M is a matching of a subset of $\mathbf{mid}(e) \backslash X$, and $cp(G_e, X, M) \geqslant \ell\}$. We observe that there exist ℓ_0 pairwise vertex-disjoint cycles in G if and only if $(\emptyset, \emptyset, \ell_0) \in \mathcal{R}_{e_r}$. We should now compute $\mathcal{R}_{e_r}$. If e is a leaf then $G_e = (\{x, y\}, \{(x, y)\})$ and $\mathcal{R}_e = \{(\emptyset, \emptyset, 0), (\emptyset, \{(x, y)\}, 0)\}$. Otherwise, let e_1 and e_2 be the two children of e in $E(T)$. $\mathcal{R}_e$ is the set of all triples (X, M, ℓ) such that there exist $(S_1, S_2) = ((X_1, M_1, \ell_1), (X_2, M_2, \ell_2)) \in \mathcal{R}_{e_1} \times \mathcal{R}_{e_2}$ such that $M \subseteq ((V[M_1] \cup V[M_2]) \cap (\mathbf{mid}(e) \backslash X))^2$, $G[S_1, S_2]$ is defined, all vertices in $\mathbf{mid}(e)$ of degree at least two in $G[S_1, S_2]$ are in X, and we can find in $G[S_1, S_2]$ ℓ_3 cycles and a path $x \ldots y$ for each $(x, y) \in M$ such that $\min(\ell_1 + \ell_2 + \ell_3, \ell_0) \geqslant \ell$.

Note that $G[S_1, S_2]$ is a minor of G so $G[S_1, S_2]$ is also planar. As we have considered a sc-decomposition and all the paths we consider in $G[S_1, S_2]$ are pairwise vertex-disjoint, since each vertex has degree at most two, the maximum number of distinct matchings M is bounded by the number of non-crossing matchings on $|\mathbf{mid}(e)|$ elements, which is at most $2^{|\mathbf{mid}(e)|}$. As we have at most $3^{|\mathbf{mid}(e)|}$ choices for X and $V[M]$, it follows that for each $e \in E(T)$, $|\mathcal{R}_e| \leqslant 6^{|\mathbf{mid}(e)|} \cdot \ell_0$. As for each $e \in E(T)$ such that e is not a leaf, we have to merge the tables of the two children e_1 and e_2 of e, this algorithm can check in time $O(36^{\mathbf{bw}} \cdot \ell_0^2 \cdot |V(G)|)$ whether G contains at least ℓ_0 vertex-disjoint cycles. We note that the constant can probably be optimized, for example by using fast matrix multiplication, but this is outside of the scope of this paper.

D Proof of Claim 1

In this proof, we say that a cycle C *kills* an another cycle C' if, for any set S of vertex-disjoint cycles containing C, $(S \backslash \{C\}) \cup \{C'\}$ is also a set of vertex-disjoint cycles. When dealing with a gadget F, we say that a cycle intersecting F is *internal* if it contains only vertices in F, and *external* otherwise.

First note that any internal cycle in an expel or a double-expel gadget should use both vertices v and v'. Also note that if some external cycle in an expel or a double-expel gadget uses the vertex v or v' of an expel or a double-expel gadget, then it also uses the vertex u (or u and u''), and then we are not able to find an internal cycle anymore. Therefore, any external cycle containing v or v' kills the cycle on the set of vertices $\{u, v, v'\}$ or $\{u', u'', v', v''\}$.

Note that if an external cycle of a path-crossing gadget turns inside it, then without loss of generality it uses a path of the form $pc_1, w_{1,1}, w_{1,2}, w_0, w_{2,2}, w_{2,1}, pc_2$ inside the path-crossing gadget. This external cycle kills the cycle inside the expel gadget between $w_{1,1}$ and $w_{2,2}$. Moreover, note that another disjoint external cycle turning in the same path-crossing gadget kills another internal cycle in the path-crossing gadget, namely the one inside the expel gadget between $w_{3,1}$ and $w_{4,2}$.

Let C be a cycle in H that is not entirely contained in only one expel, double-expel, or SC$_i$-gadget. Because of the previous remarks, we have that C cannot turn in two different path-crossing gadgets, and that if it does *not* turn in any path-crossing gadget, then by construction it uses at least two expel or double-expel gadgets and kills their internal cycles. In both configurations, adding C to the solution decreases the number of vertex-disjoint cycles that we can find in H.

The only remaining choice for C is to turn exactly once in one path-crossing gadget. If it happens inside a bifurcate gadget, then C uses vertices of two expel gadgets, namely $expel_1$ and $expel_2$, corresponding to two different colors. The only way to connect vertices corresponding to different colors outside the path-crossing gadget is by using an SC_i-gadget. So either C kills the cycles of $expel_1$ and $expel_2$, or it may also use a path leading to an edge gadget. If C turns in a path-crossing gadget inside an edge gadget, then the analysis is similar, but there is an extra case where the edge gadget representing the edge between v_i and v_j is directly plugged into the SC_j-gadget. In this case, note that none of the vertices a_i, b_i, c_i can be a selected vertex with the set of cycles we currently ask for, and therefore in order to allow it we need to decrease the number of cycles in the solution.

E Other Problems of Type 2

We can provide other examples of problems of Type 2. This is the case, for instance, of Cycle Cover, for which the lower bound has been proved in [4], and the upper bound can be proved similarly to Lemma 1.

Other problems of Type 2 are those where one wants to *maximize* the number of connected components induced by the vertices in a solution. It has been proved in [4] that Maximally Disconnected Dominating Set cannot be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ unless the ETH fails. Again, the upper bound can be proved similarly to Lemma 1. We can define more problems of this flavor, such as the following one.

> Maximally Disconnected Feedback Vertex Set
> **Input:** A graph $G = (V, E)$ and two integers ℓ and r.
> **Parameter:** The treewidth $\mathbf{tw}$ of G.
> **Question:** Does G contain a feedback vertex set of size at most ℓ that induces at least r connected components?

The following lemma can be proved by using the reduction given in [4] for Maximally Disconnected Dominating Set, just by appropriately redefining the so-called *force* and *one-in-many* gadgets.

Lemma 3. Maximally Disconnected Feedback Vertex Set *cannot be solved in time* $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ *unless the ETH fails.*

And again, the following lemma can be proved using standard dynamic programming techniques.

Lemma 4. Maximally Disconnected Feedback Vertex Set *can be solved in time* $2^{O(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$*, and* Planar Maximally Disconnected Feedback Vertex Set *can be solved in time* $2^{O(\mathbf{tw})} \cdot n^{O(1)}$*.*

F Proof of Lemma 2

Again, we prove the lemma using branch-decomposition, which will lead the same asymptotic upper bounds in terms of the treewidth. Let G be a colored graph and let $\gamma : V(G) \to \{0, \dots, \mathbf{tw}\}$ be a coloring of $V(G)$. Let $\{\mathcal{N}_i = \{s_i, t_i\}\}_{i \in [m]}$ be the endvertices of the m paths we are looking for, and let (T, μ) a branch-decomposition of G of width $\mathbf{bw} = \mathbf{bw}(G)$. As in [9], we root T by arbitrarily choosing an edge e and subdivide it by inserting a new node s. Let e' and e'' be the new edges and set $\mathbf{mid}(e') = \mathbf{mid}(e'') = \mathbf{mid}(e)$. We create a new root node r, connect it to s by an edge e_r, and set $\mathbf{mid}(e_r) = \emptyset$. The root e_r is not considered as a leaf.

Let now e be an edge of T, let $X, P \subseteq \mathbf{mid}(e)$ with $X \cap P = \emptyset$, and let M, L be two disjoint matchings of $\mathbf{mid}(e) \backslash (X \cup P)$. Let $\gamma_0 : P \cup V[M] \cup V[L] \to \{0, \dots, \mathbf{tw}\}$ be a color function, and let $\varphi : P \to [m]$ be an injective function. Intuitively, we want to keep track of the (partial) paths inside G_e, and to this end P will correspond to the virtual sources of terminals, M to the pairs of virtual sources to be linked by a path, L to pairs of vertices $\{x, y\}$ such that there is a path in G_e linking x and y, and X to vertices that are already inside a path or that are both an endpoint and a terminal. We say that $mdp(G_e, \mathbf{mid}(e), X, P, M, L, \gamma_0, \varphi) = true$ if the following conditions are fulfilled:

- For all $\{s_i, t_i\}$ in $\mathcal{N} \cap V(G_e)^2$,
  - There exists a monochromatic path $s_i \dots t_i$ in G_e, or
  - There exist $\{s_i', t_i'\} \in M$ and two monochromatic paths in G_e $s_i \dots s_i'$ colored $\gamma_0(s_i')$ and $t_i \dots t_i'$ colored $\gamma_0(t_i')$ with $\gamma_0(s_i') \equiv \gamma_0(t_i')$.
- For all $\{s_i, t_i\}$ in $\mathcal{N}$, such that $s_i \in V(G_e)$ and $t_i \notin V(G_e)$ or vice-versa,
  - There exist $s_i' \in P$ such that $\varphi(s_i') = i$ and a monochromatic path $s_i = v_0 \dots v_k = s_i'$ colored $\gamma_0(s_i')$.
- For all $\{x_i, y_i\}$ in L,
  - There exists in G_e a monochromatic path $x_i \dots y_i$ colored $\max(\gamma_0(x_i), \gamma_0(y_i))$.
- All these paths are vertex-disjoint and all vertices in $\mathbf{mid}(e)$ with degree at least 2 are in X.

Let $S_1 = (X_1, P_1, M_1, L_1, \gamma_1, \varphi_1)$ and $S_2 = (X_2, P_2, M_2, L_2, \gamma_2, \varphi_2)$ with $X_1, X_2, P_1, P_2, \dots$ defined as above. We define $G[S_1] = (P_1 \cup V[M_1] \cup V[L_1], \{\{x, y\} \in L_1\})$ and colored by γ_1, and we define $G[S_2]$ analogously. We say that $G[S_1, S_2]$ is *defined* if for all $x \in V(G[S_1]) \cap V(G[S_2])$, $\gamma_1(x) \equiv \gamma_2(x)$, $X_1 \cap V(G[S_2]) = X_2 \cap V(G[S_1]) = X_1 \cap X_2 = \emptyset$, and we define $G[S_1, S_2] = G[S_1] \cup G[S_2]$ and colored by γ_{12} such that for all $x \in V(G[S_1, S_2])$, $\gamma_{12} = \max(\gamma_1(x), \gamma_2(x))$. Otherwise, we say that $G[S_1, S_2]$ is *undefined*.

For each $e \in E(T)$, we define $\mathcal{R}_e = \{(X, P, M, L, \gamma, \varphi) | X \subseteq \mathbf{mid}(e), P \subseteq \mathbf{mid}(e)$, $X \cap P = \emptyset$, M and L are disjoint matchings on $\mathbf{mid}(e) \backslash (X \cup P)$, $V[M] \cap V[L] = \emptyset$ and $mdp(G_e, \mathbf{mid}(e), X, P, M, L, \gamma, \varphi) = true$. We want to know whether $(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \in \mathcal{R}_{e_r}$. For each $e \in E(T)$, we can compute $\mathcal{R}_e$ as follows:

- if e is a leaf, then $G_e = (\{x, y\}, \{(x, y)\}$, and
  - if $\{x, y\} \in \mathcal{N}$, then $\mathcal{R}_e = \{(\{x, y\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)\}$.
  - if $x \in \mathcal{N}_i, y \in \mathcal{N}_j$, $i \neq j$, then $\mathcal{R}_e = \{(\emptyset, \{x, y\}, \emptyset, \emptyset, \{(x, \gamma(x)), (y, \gamma(y))\}, \{(x, i), (y, j)\})\}$.
  - if $x \in \mathcal{N}_i$ and $\forall j \in [m], y \notin \mathcal{N}_j$ and $\gamma(x) \not\equiv \gamma(y)$, then $\mathcal{R}_e = \{(\emptyset, \{x\}, \emptyset, \emptyset, \{(x, \gamma(x)\}, \{(x, i)\})\}$.
  - if $x \in \mathcal{N}_i$ and $\forall j \in [m], y \notin \mathcal{N}_j$ and $\gamma(x) \equiv \gamma(y)$, then $\mathcal{R}_e = \{(\emptyset, \{x\}, \emptyset, \emptyset, \{(x, \gamma(x)\}, \{(x, i)\}), (\{x\}, \{y\}, \emptyset, \emptyset, \{(y, \max(\gamma(x), \gamma(y)))\}, \{(y, i)\})\}$.
- if e is not a leaf, let e_1 and e_2 be the two children of e in $E(T)$. We construct $\mathcal{R}_e$ as the set of all 6-tuples $(X, P, M, L, \gamma_0, \varphi)$ such that there exist $S_1 = (X_1, P_1, M_1, L_1, \gamma_1, \varphi_1) \in \mathcal{R}_{e_1}$ and $S_2 = (X_2, P_2, M_2, L_2, \gamma_2, \varphi_2) \in \mathcal{R}_{e_2}$ fulfilling the following properties:

- $H = G[S_1, S_2]$ is defined;
- For all $\{x_i, y_i\} \in L$, there exists a monochromatic path $x_i \dots y_i$ in H and we have $\gamma_0(x_i) = \gamma_0(y_i) = \gamma_{12}(x_i \dots y_i)$;
- All vertices in $\mathbf{mid}(e)$ of degree at least 2 in $G[S_1, S_2]$ are in X;
- For all $\{v, w\} \in M_i$, $i \in \{1, 2\}$, there is a monochromatic color-compatible path from v to w in $G[S_1, S_2]$ or two vertices $\{v', w'\} \in M$, and two monochromatic color-compatible paths $v \dots v'$ and $w \dots w'$ with $\gamma_0(v') = \gamma_0(w') = \max(\gamma_{12}(v \dots v'), \gamma_{12}(w \dots w'))$;
- For all $i \in \{1, 2\}$ an for all v in P_i, there exist $w \in P$ and a monochromatic color-compatible path $v \dots w$, or there exist $w \in P_{3-i}$ such that $\varphi_i(v) = \varphi_{3-i}(w)$ and a monochromatic path $v \dots w$ such that $\gamma_0(w) = \gamma_{12}(v \dots w)$;
- All these paths are pairwise vertex-disjoint.

As in the graph $G[S_1, S_2]$ by construction all vertices have degree at most two, we can easily check all the previous properties in polynomial time, as we just have to compare two sets or traverse a path in $G[S_1, S_2]$ to verify each property. Therefore, we can compute each element of $\mathcal{R}_e$ in time poly($\mathbf{mid}(e)$). As $(X, P, V[M], V[L])$ forms a partition of a subset of $\mathbf{mid}(e)$, there are at most $5^{\mathbf{mid}(e)}$ such 4-tuples. There are at most $\mathbf{tw}+1$ colors and at most $(\mathbf{tw}+1)^{\mathbf{mid}(e)}$ choices for γ_0. As $|\{\varphi(x) | x \in P\}| \leqslant |P| \leqslant \mathbf{mid}(e)$, there are at most $\mathbf{mid}(e)^{\mathbf{mid}(e)}$ possible different color functions φ. As $\mathbf{bw} - 1 \leqslant \mathbf{tw}$ we have that for all e in $E(T)$, $|\mathbf{mid}(e)| \leqslant \mathbf{tw}+1$, hence for all e in $E(T)$, $|\mathcal{R}_e| \leqslant 5^{\mathbf{tw}+1} \cdot (\mathbf{tw}+1)^{2(\mathbf{tw}+1)}$. As for each $e \in E(T)$ such that e is not a leaf, we have to merge the tables of the two children e_1 and e_2 of e, the above dynamic programming algorithm can solve Monochromatic Disjoint Paths in time $O(25^{\mathbf{tw}+1} \cdot (\mathbf{tw}+1)^{4(\mathbf{tw}+1)} \cdot |V(G)|)$. Again, we note that the constant can probably be optimized by using fast matrix multiplication.

G Fig. 13 in the Proof of Theorem 4

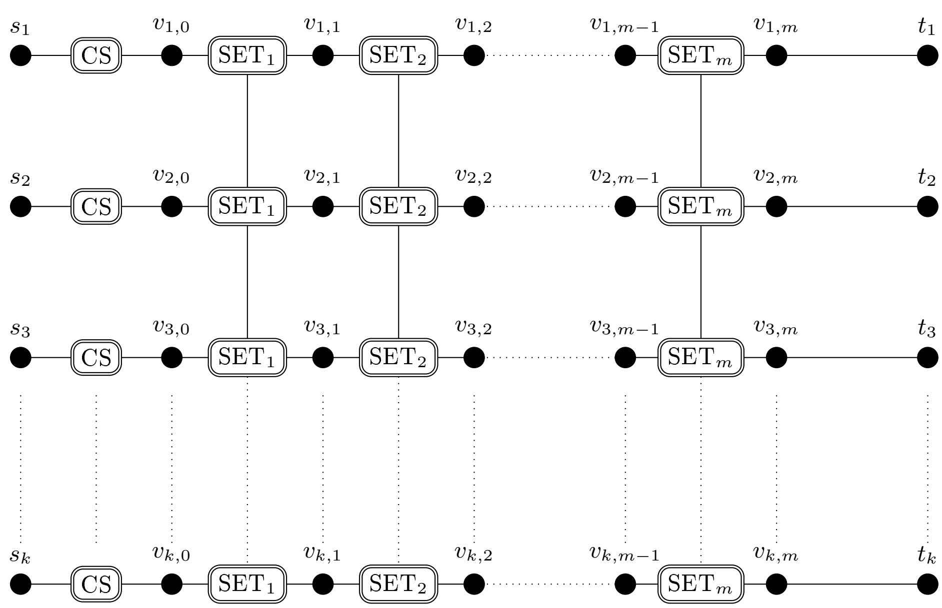


Fig. 13. Final graph G in the reduction of Theorem 4.

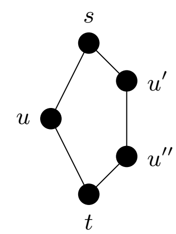


Fig. 14. The double-expel gadget.

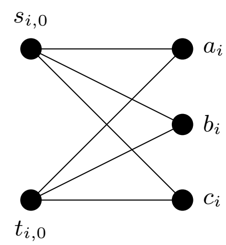


Fig. 15. The SC_i-gadget: To keep planarity, there is a path-crossing gadget in each edge intersection.

H Description of the Graph G in the Proof of Theorem 4

Formally, the graph we obtain is $G = (V, E)$, where $V = \{s_r | r \in [k]\} \cup \{t_r | r \in [k]\} \cup \{v_{r,i} | r \in [k], i \in \{0, m\}\} \cup \{u_{r,c} | r \in [k], c \in [k]\} \cup (\{w_{r,i,b} | r \in [k], i \in [m], i \in \{1,2\}\} \setminus \{w_{r,i,b} | i \in [m], (r,b) \in \{(1,1),(k,2)\}\}) \cup \{s_{r,i} | r \in [k-1], i \in [m]\} \cup \{t_{r,i} | r \in [k-1], i \in [m]\} \cup \{a_{r,i} | \exists c \in [k], (r,c) \in S_i$ and $E = \{\{s_r, u_{r,c}\} \in V^2 | r \in [k], c \in [k]\} \cup \{\{u_{r,c}, v_{r,0}\} \in V^2 | r \in [k], c \in [k]\} \cup \{\{v_{r,i-1}, w_{r,i,b}\} \in V^2 | r \in [k], i \in [m], b \in \{1,2\}\} \cup \{\{w_{r,i,b}, v_{r,i}\} \in V^2 | r \in [k], i \in [m], b \in \{1,2\}\} \cup \{\{v_{r,i-1}, a_{r,i}\} \in V^2 | r \in [k], i \in [m]\} \cup \{\{a_{r,i}, v_{r,i}\} \in V^2 | r \in [k], i \in [m]\} \cup \{\{v_{r,m}, t_r\} \in V^2 | r \in [k]\} \cup \{\{s_{r,i}, w_{r,i,2}\} \in V^2 | r \in [k-1], i \in [m]\} \cup \{\{s_{r,i}, w_{r+1,i,1}\} \in V^2 | r \in [k-1], i \in [m]\} \cup \{\{t_{r,i}, w_{r,i,2}\} \in V^2 | r \in [k-1], i \in [m]\} \cup \{\{t_{r,i}, w_{r+1,i,1}\} \in V^2 | r \in [k-1], i \in [m]\}$.

I Lower Bound for Planar Disjoint Paths

In this section we prove that, assuming the ETH, the Planar Disjoint Paths problem cannot be solved in time $2^{o(\mathbf{tw})} \cdot n^{O(1)}$.

Disjoint Paths **Input:** A graph $G = (V, E)$, an integer m, and a set $\mathcal{N} = \{\mathcal{N}_i = \{s_i, t_i\} | i \in [m], s_i, t_i \in V\}$. **Parameter:** The treewidth $\mathbf{tw}$ of G. **Question:** Does G contain m pairwise vertex-disjoint paths from s_i to t_i, for $i \in [m]$?

Theorem 5. Planar Disjoint paths *cannot be solved in time* $2^{o(\sqrt{n})} \cdot n^{O(1)}$ *unless the ETH fails.*

Proof: We strongly follow the proof of Theorem 2. Again, we reduce from Planar 3-Colorability where the input graph has maximum degree at most 5. Let $G = (V, E)$

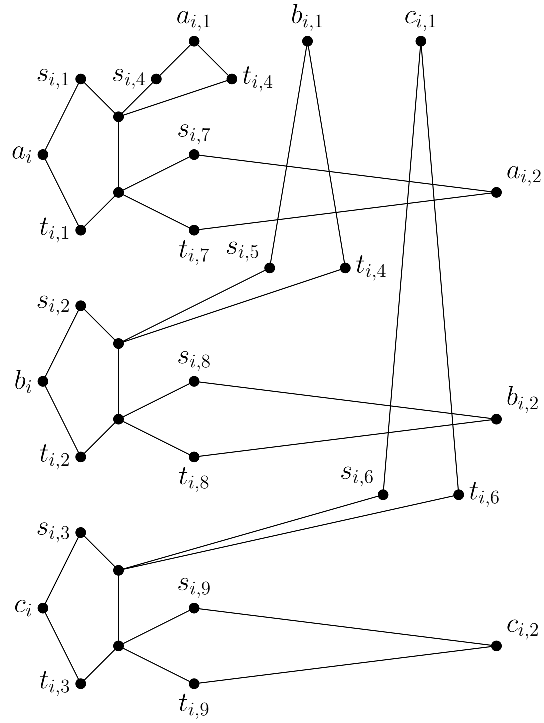


Fig. 16. Bifurcate gadget: To keep planarity, there is a path-crossing gadget in each edge intersection.

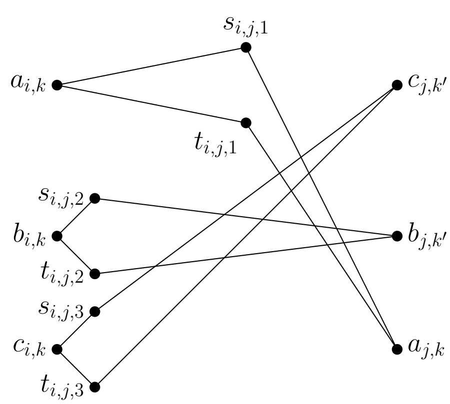


Fig. 17. Edge gadget: To keep planarity, there is a path-crossing gadget in each edge intersection.

be a planar graph with maximum degree at most 5 with $V = \{v_1, \ldots, v_n\}$. We proceed to construct a planar graph H together with a planar embedding. We construct the same graph as in the proof of Theorem 2 but where the gadgets are appropriately modified. We reuse the *expel* gadget depicted in Fig. 6 and for each expel gadget, we ask for a path between s and t. We redefine the *double-expel* gadget as depicted in Fig. 14 and for each double-expel gadget, we ask for a path between s and t. We reuse the *path-crossing* gadget depicted in Fig. 3 and only ask for the paths contained in the expel gadgets. We can now redefine the SC_i-gadget depicted in Fig. 15, where each edge intersection is replaced with a path-crossing gadget. For each SC_i-gadget we ask for a path between $s_{i,0}$ and $t_{i,0}$. We also redefine the *bifurcate* gadget as depicted in Fig. 16, and for each bifurcate gadget, we ask for a path between $s_{i,k}$ and $t_{i,k}$ for $k \in [9]$. Finally, we redefine the *edge* gadget as depicted in Fig. 17, and for each edge gadget we ask for a path between $s_{i,j,k}$ and $t_{i,j,k}$ for $k \in [3]$. This completes the construction of the planar graph H. It can be easily checked that these gadgets preserve the same properties as the corresponding ones in the proof of Theorem 2. Moreover, it is also easy to see that a path in a solution in H cannot turn in a path-crossing gadget.

Given a solution of PLANAR DISJOINT PATHS in H, for each $i \in [n]$ the selection of a cycle in the SC_i-gadget selects a color for v_i, that can be any common color in all color outputs of v_i, and the edge gadgets ensure that two adjacent vertices are in two different color classes. So in this way we obtain a solution of PLANAR 3-COLORABILITY in G.

Conversely, given a solution of PLANAR 3-COLORABILITY in G, it defines a color output for $\{a_i, b_i, c_i\}$ for $i \in [n]$. Therefore, we select in the SC_i-gadget the path that uses the vertex in $\{a_i, b_i, c_i\}$ corresponding to the color of v_i. In each bifurcate gadget, we choose the paths that use the vertices in $\{a_{i,1}, b_{i,1}, c_{i,1}, a_{i,2}, b_{i,2}, c_{i,2}\}$ leading to two identical color outputs that coincide with the color output of $\{a_i, b_i, c_i\}$. This choice satisfies the property that the color output of $\{a_i, b_i, v_i\}$ is contained in the color outputs of $\{a_{i,1}, b_{i,1}, c_{i,1}\}$ and $\{a_{i,2}, b_{i,2}, c_{i,2}\}$, and leaves as many free vertices as possible for other cycles in other gadgets. Inside each edge gadget representing $\{v_i, v_j\} \in E$, we select the paths that are allowed by the free vertices. We complete our path selection by selecting a free path in each expel gadget contained in the path-crossing gadget.

As the degree of each vertex in G is bounded by 5, the number of gadgets we introduce for each $v_i \in V(G)$ in order to construct H is also bounded by a constant, so the total number of vertices of H is linear in the number of vertices of G. Therefore, if we could solve PLANAR DISJOINT PATHS in time $2^{o(\sqrt{n})} \cdot n^{O(1)}$, then we could also solve PLANAR 3-COLORABILITY in time $2^{o(\sqrt{n})} \cdot n^{O(1)}$, which is impossible by Theorem 1 unless the ETH fails. The theorem follows. □

From Theorem 5 we obtain the following corollary.

Corollary 2. PLANAR DISJOINT PATHS *cannot be solved in time* $2^{o(\mathbf{tw})} \cdot n^{O(1)}$ *unless the ETH fails.*

J Lower Bound for Planar Subgraph Isomorphism

Using ideas similar to those in the reduction for MONOCHROMATIC DISJOINT PATHS in Theorem 4, we can prove that other planar problems cannot be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})}$ unless the ETH fails. Intuitively, we can prove lower bounds along the same ideas for problems where some information can be carried by paths that belong to the solution. For example, this is the case of PLANAR SUBGRAPH ISOMORPHISM.

PLANAR SUBGRAPH ISOMORPHISM
Input: Two planar graphs G and H.
Parameter: The treewidth $\mathbf{tw}$ of (G).
Question: Does G contain a subgraph isomorphic to H?

Theorem 6. PLANAR SUBGRAPH ISOMORPHISM *cannot be solved in time* $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{O(1)}$ *unless the ETH fails.*

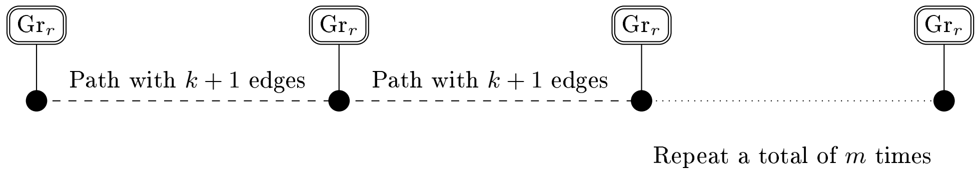


Fig. 18. Req_r with $m+1$ elements Gr$_r$.

Fig. 19. Expel gadget.

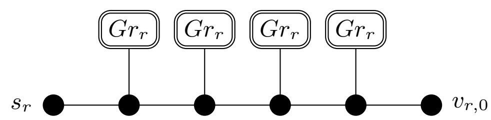


Fig. 20. The color-selection gadget.

Proof: We reduce again from $k \times k$-Hitting Set. Let $k \geqslant 3$ be an integer and $S_1, S_2, \ldots, S_m \subseteq [k] \times [k]$ such that each set contains at most one element from each row of $[k] \times [k]$. We proceed to construct a graph G similar to the one for Monochromatic Disjoint Paths, and we simultaneously construct the graph H. In this section, each connected component of H is called a *request*, so H will consist of the union of all the constructed requests. We now explain how we modify the expel, the color-selection, and the set gadgets.

As in Monochromatic Disjoint Paths, we create a gadget for each row $\{r\} \times [k]$ and make for each of these gadgets a request Req_r, depicted in Fig. 18, that chooses a path and simulates a color for each row. We will again add an appropriate set gadget ensuring that $S \cap S_i \neq \emptyset$ for each $i \in [m]$. For each $r \in [k]$, the request Req_r simulates the path from s_r to t_r that appears in Monochromatic Disjoint Paths. The way it is inserted in the color-selection gadget simulates the color of the path $s_r \ldots t_r$ and determines the paths it can use in the set gadgets, similarly to the reduction for Monochromatic Disjoint Paths.

More precisely, we define $k+1$ graphs that do not appear anywhere else in the graph as a subgraph. For instance, we can take a grid of size $k+2$ in which we remove a different vertex every time. Let $Gr_1, Gr_2, \ldots, Gr_k$ and Gr_g be these grids. Each such modified grid Gr_i, $i \in [k]$, identifies the subgraph corresponding to the row i, while Gr_g identifies the expel gadget requests. We redefine the *expel* gadget as depicted in Fig. 19 and the *color-selection* gadget as depicted in Fig. 20. We also redefine the *set* gadget as depicted in Fig. 21, similarly to Monochromatic Disjoint Paths but with the new expel gadget and a way to simulate in Planar Subgraph Isomorphism the color we had in Monochromatic Disjoint Paths.

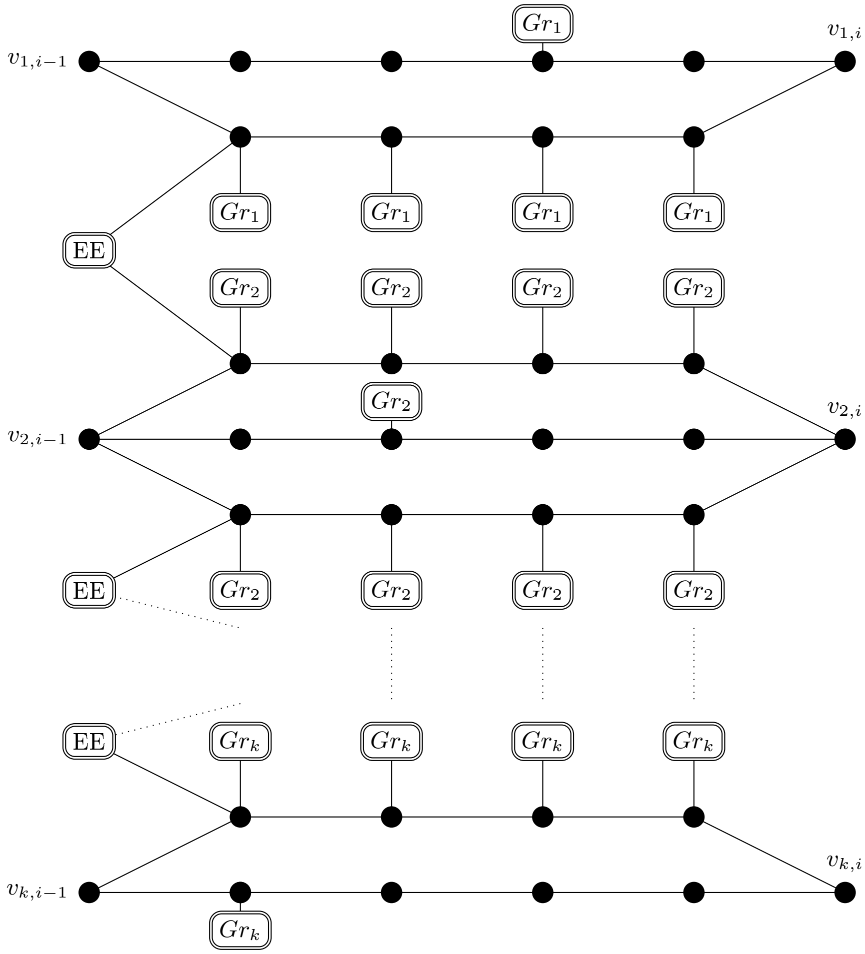


Fig. 21. Set gadget with $(1,3) \in S_1$, $(2,2) \in S_2$, $\ldots$, $(k,1) \in S_k$.

The final graph we construct has the same shape depicted in Fig. 13 but with the new gadgets instead. We define the planar graph H to be the union of the $m \cdot (k-1)$ expel gadget requests shown in Fig. 19 and the request Req_r for each $r \in [k]$.

All we have to specify is that we can use the path in the set gadget that simulates the colored path in the reduction of Planar Monochromatic Disjoint Paths only if we choose the corresponding color in the color-selection gadget. This property is given by the fact that we look for a path where the constructed grids are spaced by the same number of vertices each time, and therefore the color selected in the color-selection gadget is preserved in each set gadget.

If S is a solution of $k \times k$-Hitting Set, by finding the request Req_i that simulates the corresponding color we can construct solution to Planar Subgraph Isomorphism, similarly to the way we found a solution of Monochromatic Disjoint Paths. Conversely, for the same reason in the proof for Monochromatic Disjoint Paths, a solution of Planar Subgraph Isomorphism also defines a solution S of $k \times k$-Hitting Set. $\square$