Complexity Framework For Forbidden Subgraphs

Matthew Johnson[1], Barnaby Martin[1], Jelle J. Oostveen[2], Sukanya Pandey[2], Daniël Paulusma[1], Siani Smith[1], and Erik Jan van Leeuwen[2]

[1] Department of Computer Science, Durham University, Durham, UK, {matthew.johnson2,barnaby.d.martin,daniel.paulusma,siani.smith}@durham.ac.uk
[2] Department of Information and Computing Sciences, Utrecht University, The Netherlands, {j.j.oostveen,s.pandey1,e.j.vanleeuwen}@uu.nl

Abstract. For any finite set $\mathcal{H} = \{H_1, \ldots, H_p\}$ of graphs, a graph is $\mathcal{H}$-subgraph-free if it does not contain any of $H_1, \ldots, H_p$ as a subgraph. Similar to known meta-classifications for the minor and topological minor relations, we give a meta-classification for the subgraph relation. Our framework classifies if problems are "efficiently solvable" or "computationally hard" for $\mathcal{H}$-subgraph-free graphs. The conditions are that the problem should be efficiently solvable on graphs of bounded treewidth, computationally hard on subcubic graphs, and computational hardness is preserved under edge subdivision. We show that all problems satisfying these conditions are efficiently solvable if $\mathcal{H}$ contains a disjoint union of one or more paths and subdivided claws, and are computationally hard otherwise. To illustrate the broad applicability of our framework, we study partitioning, covering and packing problems, network design problems and width parameter problems. We apply the framework to obtain a dichotomy between polynomial-time solvability and NP-completeness. For other problems we obtain a dichotomy between almost-linear-time solvability and having no subquadratic-time algorithm (conditioned on some hardness hypotheses). Along the way we unify and strengthen known results from the literature.

Keywords: forbidden subgraph; complexity dichotomy; treewidth

1 Introduction

Algorithmic meta-theorems are general algorithmic results applying to a whole range of problems, rather than just a single problem alone [58]. An *algorithmic meta-theorem* is a statement saying that all problems sharing some property or properties P, restricted to a class of inputs I, can be solved efficiently by a certain form of algorithm.

Probably the most famous algorithmic meta-theorem is that of Courcelle [24], which proves that every graph property expressible in monadic second-order logic is decidable in linear time if restricted to graphs of bounded *treewidth* (see Section 2 for a definition of treewidth). Another example is that of Seese [72], which proves that every graph property expressible in first-order logic is decidable in linear time when restricted to graphs of bounded degree. A third example comes from Dawar et al. [28], who proved that every first-order definable optimisation problem admits a polynomial-time approximation scheme on any class of graphs excluding at least one minor. There is a wealth of further algorithmic meta-theorems (see, for example, [14,29,36]), many of which combine structural graph theory (e.g. from graph minors) with logic formulations or other broad problem properties (such as bidimensionality).

An extension of an algorithmic meta-theorem can produce a so-called *algorithmic meta-classification.* This is a general statement saying that all problems that share some property or properties P admit, over some classes of input restrictions I, a classification according to whether or not they have property S. If the input-restricted class has property S, then this class is efficiently solvable; otherwise it is computationally hard.

Algorithmic meta-classifications are less common than algorithmic meta-theorems, but let us mention two famous results. Grohe [45] proved that there is a polynomial-time algorithm for finite-domain constraint satisfaction problems whose left-hand input structure is restricted to $\mathcal{C}$ if and only if $\mathcal{C}$ has bounded treewidth (assuming W[1] $\neq$ FPT). Bulatov [20] and Zhuk [83] proved that every finite-domain CSP(H) is either polynomial-time solvable or NP-complete, omitting any Ladner-like complexities in between.

In this paper, we pioneer an algorithmic meta-classification for graph problems where the class of input restrictions comes from omitting each of a finite set $\mathcal{H}$ of graphs *as a subgraph*.

1.1 Forbidding Some Graph Pattern

A graph G contains a graph H as a *subgraph* if G can be modified to H by a sequence of vertex deletions and edge deletions; if not, then G is *H-subgraph-free*. A graph G contains H as an *induced subgraph* if G can be modified to H by a sequence of only vertex deletions; if not, then G is *H-free*.

The *contraction* of an edge $e = uv$ in a graph replaces u and v by a new vertex that is made adjacent precisely to the former neighbours of u and v in G (without creating multiple edges). If v had degree 2 and its two neighbours in G are non-adjacent, then we also say that we *dissolved* v. A graph G contains H as a *topological minor* (or as a *subdivision*) if G can be modified to H by a sequence of vertex deletions, vertex dissolutions and edge deletions; if not, then G is *H-topological-minor-free*. A graph G contains H as a *minor* if G can be modified to H by a sequence of vertex deletions, edge deletions and edge contractions; if not, then G is *H-minor-free*.

For a set $\mathcal{H}$ of graphs, a graph G is *$\mathcal{H}$-subgraph-free* if G is H-subgraph-free for every $H \in \mathcal{H}$. If $\mathcal{H} = \{H_1, \dots, H_p\}$ for some integer $p \geq 1$, we also say that G is *$(H_1, \dots, H_p)$-subgraph-free*. Graph classes closed under edge deletion are also called *monotone* [5,18,57]. We also define the analogous notions of being $\mathcal{H}$-free, $\mathcal{H}$-topological-minor-free and $\mathcal{H}$-minor-free. A class of $\mathcal{H}$-free graphs is also called *hereditary*.

We now make the following observation that holds for every class of graphs $\mathcal{H}$ and that shows how the four containment relations relate to each other:

$\mathcal{H}$-minor-free graphs $\subseteq$ $\mathcal{H}$-topological-minor-free graphs $\subseteq$ $\mathcal{H}$-subgraph-free graphs $\subseteq$ $\mathcal{H}$-free graphs.

The minor and topological minor relations are well understood due to the Robertson-Seymour Theorem. This is shown by two well-known meta-classifications, which are a consequence of a classic result of [70]; we refer to Appendix A for some more details, but see also e.g. [57]. Both theorems hold for infinite sets $\mathcal{H}$ as well. They distinguish between "efficiently solvable" and "computationally hard". We let these notions depend on context. For example, efficiently solvable and computationally hard could mean being solvable in polynomial time and being NP-complete, respectively. A graph is *subcubic* if every vertex has maximum degree at most 3.

Theorem 1. *Let Π be a problem that is computationally hard on planar graphs, but efficiently solvable for every graph class of bounded treewidth. For any set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-minor-free graphs is efficiently solvable if $\mathcal{H}$ contains a planar graph and is computationally hard otherwise.*

Theorem 2. *Let Π be a problem that is computationally hard on planar subcubic graphs, but efficiently solvable for every graph class of bounded treewidth. For any set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-topological-minor-free graphs is efficiently solvable if $\mathcal{H}$ contains a planar subcubic graph and is computationally hard otherwise.*

In our paper, we will discuss many problems that satisfy the conditions of Theorems 1 and 2. We refer, for example, to [37,66] for a number of problems that satisfy the conditions of Theorem 2, and thus also of Theorem 1, and that are NP-complete even for planar subcubic graphs of high girth. However, in contrast to the minor and topological minor relations, even algorithmic meta-theorems are unknown for the induced subgraph relation, not even for a single forbidden graph H (complexity dichotomies for H-free graphs are only known for specific problems, see e.g. [13,43,53,54]). In contrast to the family of minor-closed graph classes, the family of hereditary graph classes contains infinite descending chains of graph classes. This makes the task of finding algorithmic meta-theorems much harder. To deal with this obstacle, Alekseev [4] introduced the notion of a boundary graph class; see, e.g., [5,57,66] for more details on this notion.

In this paper we focus on the hitherto less-studied "in-between" relation: the subgraph relation. The subgraph relation also represents many rich graph classes. Here are some examples:

1. The class of graphs of maximum degree r coincides with the class of $K_{1,r+1}$-subgraph-free graphs; here, $K_{1,r}$ denotes the $(r+1)$-vertex star.
2. A graph class $\mathcal{G}$ has bounded tree-depth if and only if there is an integer $r \geq 1$ such that every graph in $\mathcal{G}$ is P_r-subgraph-free [67]; here, P_r denotes the path on r vertices.
3. A graph is H-subgraph-free if and only if it is $\mathcal{H}$-free where $\mathcal{H}$ consists of all graphs containing H as a spanning subgraph.

Note that the last example implies that the classes of H-free graphs and H-subgraph-free graphs coincide if and only if $H = K_r$ for some integer $r \geq 1$ (the graph K_r is the complete graph on r vertices).

1.2 Known Complexity Classifications for the Subgraph Relation

Despite capturing natural graph classes, the subgraph relation has been significantly less studied in the context of algorithms than the other two relations. Kamiński [53] gave a complexity classification for Max-Cut restricted to $\mathcal{H}$-subgraph-free graphs, where $\mathcal{H}$ is any finite set of graphs. Twenty years earlier, Alekseev and Korobitsyn [6] did the same for Independent Set, Dominating Set and Long Path; see [44] for a short, alternative proof, similar to the one of [53] for Max-Cut, for the classification for Independent Set for H-subgraph-free graphs. In [43] the complexity of List Colouring for $\mathcal{H}$-subgraph-free graphs has been determined for every finite set of graphs $\mathcal{H}$. However, even for a classical problem such as Colouring, a complete complexity classification for H-subgraph-free graphs is far from settled [44]. More recently, Bodlaender et al. [16] determined the computational complexity of Subgraph Isomorphism for H-subgraph-free graphs for all connected graphs H except one (namely when $H = P_5$).

We now discuss our algorithmic meta-classification for $\mathcal{H}$-subgraph-free graph classes where $\mathcal{H}$ is any finite set of graphs. It is wide reaching and includes the aforementioned known classifications for Dominating Set, Independent Set, List Colouring, Long Path and Max Cut.

1.3 The Framework

We first note that jumps in complexity can be extreme and unexpected. For example, there exist problems that are PSPACE-complete in general but constant-time solvable for every $\mathcal{H}$-free graph class [62] and thus for every $\mathcal{H}$-subgraph-free graph class, where $\mathcal{H}$ is any (possibly infinite) nonempty set of graphs. Another example is the Clique problem: determine the size of a maximum clique in a graph. This problem is well-known to be NP-hard (see [41]). Now, let $\mathcal{H}$ be *any*, possibly infinite, set of graphs and h be the number of vertices of a smallest graph in $\mathcal{H}$. As every clique in any $\mathcal{H}$-subgraph-free graph has size at most $h-1$, Clique is polynomial-time solvable for $\mathcal{H}$-subgraph-free graphs. These examples show that a meta-classification for the subgraph relation can expose interesting problem behaviour.

Before we define our framework, we first give some terminology. A class of graphs has bounded treewidth if there is a constant c such that every graph in it has treewidth at most c. Recall that a graph is subcubic if every vertex has degree at most 3. The *subdivision* of an edge $e = uv$ of a graph replaces e by a new vertex w and edges uw and wv. For an integer $k \geq 1$, the *k-subdivision* of a graph G is the graph obtained from G after subdividing each edge of G exactly k times. For a graph class $\mathcal{G}$ and an integer k, let $\mathcal{G}^k$ consist of the k-subdivisions of the graphs in $\mathcal{G}$. The 4-vertex star is known as the *claw*. A *subdivided* claw is obtained from a claw after subdividing each of its edges zero or more times. The *disjoint union* $G_1 + G_2$ of two vertex-disjoint graphs G_1 and G_2 is the graph $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. The set $\mathcal{S}$ consists of all non-empty disjoint unions of a set of zero or more subdivided claws and paths; see Fig. 1 for an example of a graph that belongs to $\mathcal{S}$.
A graph problem Π is computationally hard *under edge subdivision of subcubic graphs* if there is a $k \geq 1$ such that: if Π is computationally hard for the class $\mathcal{G}$ of subcubic graphs, then Π is computationally hard for $\mathcal{G}^{kp}$ for every $p \geq 1$; see Section 3.2 for some reasons why we cannot simplify this definition.

Our framework contains every graph problem Π satisfying the following three conditions:

C1. Π is efficiently solvable for every graph class of bounded treewidth;

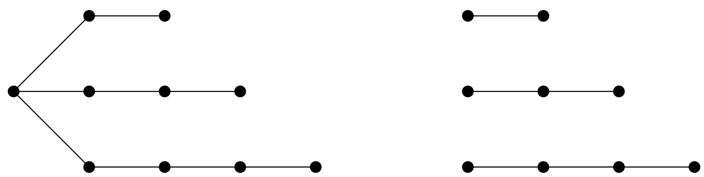

Fig. 1. An example of a graph that belongs to $\mathcal{S}$, namely the graph $S_{2,3,4} + P_2 + P_3 + P_4$.

C2. Π is computationally hard for the class of subcubic graphs; and
C3. Π is computationally hard under edge subdivision of subcubic graphs.

A problem Π that satisfies conditions C1–C3 is called a *C123-problem*. As mentioned, the notions of efficiently solvable and computational hardness depend on context. In Section 2 we show the following theorem that can be seen as the "subgraph variant" of Theorems 1 and 2.

Theorem 3. *Let Π be a C123-problem. For any finite set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-subgraph-free graphs is efficiently solvable if $\mathcal{H}$ contains a graph from $\mathcal{S}$ and computationally hard otherwise.*

The proof of Theorem 3 is simple and uses, in a more general way, the same arguments, including a well-known path-width result [9], as the proofs for Max-Cut [53] and List Colouring [43] for $\mathcal{H}$-subgraphs-free graphs (for finite $\mathcal{H}$) and Independent Set for H-subgraph-free graphs [44].[3]

In Section 3 we show that $\mathcal{H}$ cannot be allowed to have infinite size, and that in C3 we need the graph class $\mathcal{G}$ to be subcubic in order for our proof to work. In the same section we also show that there are problems that are not C123 but that still have the same dichotomy as in Theorem 3.

1.4 Alternative Characterizations

It is well known that for a set of graphs $\mathcal{H}$, a class of $\mathcal{H}$-minor-free graphs has bounded treewidth if and only if $\mathcal{H}$ contains a planar graph; a class of $\mathcal{H}$-topological-minor-free graph has bounded treewidth if and only if $\mathcal{H}$ contains a planar subcubic graph; and should $\mathcal{H}$ be finite, a class of $\mathcal{H}$-subgraph-free graphs has bounded tree-width if and only if $\mathcal{H}$ contains a graph from $\mathcal{S}$. These facts follow directly from results of Robertson and Seymour [69] (see e.g. [18,25], where this is explained with respect to the more general parameter clique-width). Hence, Theorems 1–3 imply the following alternative theorems, which suggest that boundedness of treewidth might be the underlying explanation for the polynomial-time solvability.

Corollary 1. *Let Π be a problem that is computationally hard on planar graphs, but efficiently solvable for every graph class of bounded treewidth. For any set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-minor-free graphs is efficiently solvable if the class of $\mathcal{H}$-minor-free graphs has bounded treewidth and is computationally hard otherwise.*

Corollary 2. *Let Π be a problem that is* NP*-complete on planar subcubic graphs, but efficiently solvable for every graph class of bounded treewidth. For any set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-topological-minor-free graphs is efficiently solvable if the class of $\mathcal{H}$-minor-free graphs has bounded treewidth and is computationally hard otherwise.*

Corollary 3. *Let Π be a C123-problem. For any finite set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-subgraph-free graphs is efficiently solvable if the class of $\mathcal{H}$-subgraph-free graphs has bounded treewidth and computationally hard otherwise.*

[3] The original proofs from [6] for Independent Set, Dominating Set and Long Path, restricted to $\mathcal{H}$-free graphs for finite $\mathcal{H}$, are different and do not involve any direct path-width arguments.

1.5 Consequences of Theorem 3

In Sections 4 and 5 we show a number of consequences of our meta-classification in Theorem 3. Namely, we give many C123-problems both for the distinction: polynomial-time solvable versus NP-complete, and for the distinction: almost-linear-time solvable versus quadratic-time solvable.

In Section 4, we first show that computing the path-width and treewidth of a graph are C123-problems. We also show that List Colouring is C123, and thus we obtain the classification of [43]. Moreover, we show that Independent Set (or equivalently, Vertex Cover) and Dominating Set are C123, and thus we also recover the known classifications of [6]. We do the same for a number of other covering and packing problems: (Independent) Odd Cycle Transversal, P_3-Factor and two variants of the Dominating Set problem, namely Independent Dominating Set and Edge Dominating Set; the latter is polynomially equivalent to Minimum Maximal Matching [47]. Next we prove (still in Section 4) that the following network design problems are all C123-problems: Edge/Node Steiner Tree, (Induced) Disjoint Paths, Long (Induced) Cycle, Long (Induced) Path, Max-Cut and Edge/Node Multiway Cut. Hence, we also recover the classification of [6] for Long Path restricted to $\mathcal{H}$-subgraph-free graphs. We also include a reference to a recent result of [34], in which it was shown that Perfect Matching Cut is C123.

For all the above C123 problems, Theorem 3 gives a dichotomy between polynomial-time solvability versus NP-completeness. In Section 5, we consider the polynomial-time solvable problems Diameter and Radius, which are studied in fine-grained complexity. Here, Theorem 3 gives a distinction between almost-linear-time solvability versus not having a subquadratic-time algorithm under the Orthogonal Vectors Conjecture [78] and Hitting Set Conjecture [1], respectively. The Orthogonal Vectors conjecture is implied by SETH [78] and by the Hitting Set Conjecture [1]. The latter is likely not implied by SETH; we refer to [79] for more context on both conjectures.

The above results are summarized in Table 1, which we explain further in Section 1.6.

Relation with the literature. As explained in detail in Sections 4 and 5, the fact that various problems are C123 follows from known results, or combinations of known results from the literature. This is always the case for showing C1 and often the case for C2 and C3. We prove, as new results, that List Colouring, Odd Cycle Transversal and Independent Odd Cycle Transversal are NP-complete for planar subcubic graphs, and thus satisfy C2. Moreover, for the following problems we will give explicit proofs to show that they satisfy C3: Path-Width, Tree-Width, Edge/Node Steiner Tree, Edge/Node Multiway Cut and Diameter. Our framework in fact helped us to identify these gaps in the literature.

1.6 Comparing Theorems 1–3

We make a full comparison between the minor, topological minor and subgraph frameworks; see also Table 1. We first note that every problem that is NP-complete for graphs of bounded treewidth does not satisfy any of the frameworks. An example is the aforementioned Subgraph Isomorphism problem, which is NP-complete even for input pairs (G_1, G_2) that are linear forests (see, for example, [16] for a proof) and thus have treewidth 1. As an example on the other extreme end, the Clique problem does not fall under the subgraph framework either. This is because Clique is polynomial-time solvable for $\mathcal{H}$-subgraph-free graphs for every set of graphs $\mathcal{H}$ (as we observed in Section 1.3). We also note that Clique is polynomial-time solvable for planar graphs (as every clique in a planar graph has size at most 4). Hence, it does not belong under the minor and topological minor frameworks either.

We observe that every problem that satisfies the conditions of Theorem 2 also satisfies the conditions of Theorem 1. However, there exist problems that satisfy the conditions of Theorem 1 but not those of Theorems 2 and 3. For example, 3-Colouring satisfies C1 (this even holds for List Colouring [50]), and it is NP-complete even for 4-regular planar graphs [26]. However, 3-Colouring does not satisfy the conditions of Theorems 2 and 3, as 3-Colouring is polynomial-time solvable for subcubic graphs due to Brooks' Theorem [19]. Hence, 3-Colouring does not satisfy C2 and is thus neither NP-complete for planar subcubic graphs nor a C123-problem.

We also note that, unlike Vertex Cover and Odd Cycle Transversal, the related problems Connected Vertex Cover and Feedback Vertex Set do not satisfy the conditions of Theorems 2 and 3.

graph problem	minor framework	topological minor framework	subgraph framework
Path-Width	yes	yes	yes
Tree-Width	?	?	yes
Dominating Set	yes	yes	yes
Independent Dominating Set	yes	yes	yes
Edge Dominating Set	yes	yes	yes
Independent Set	yes	yes	yes
Vertex Cover	yes	yes	yes
Connected Vertex Cover	yes	**no**	**no**
Feedback Vertex Set	yes	**no**	**no**
Independent Feedback Vertex Set	yes	**no**	**no**
Odd Cycle Transversal	yes	yes	yes
Independent Odd Cycle Transversal	yes	yes	yes
3-Colouring	yes	**no**	**no**
List Colouring	yes	yes	yes
P_3-Factor	yes	yes	yes
Edge Steiner Tree	yes	yes	yes
Node Steiner Tree	yes	yes	yes
Disjoint Paths	yes	yes	yes
Induced Disjoint Paths	yes	yes	yes
Long Cycle	yes	yes	yes
Long Induced Cycle	yes	yes	yes
Hamilton Cycle	yes	yes	**no**
Long Path	yes	yes	yes
Long Induced Path	yes	yes	yes
Hamilton Path	yes	yes	**no**
Max-Cut	**no**	**no**	yes
Edge Multiway Cut	yes	yes	yes
Node Multiway Cut	yes	yes	yes
Perfect Matching Cut	?	?	yes
Diameter	?	?	yes
Radius	?	?	yes
Subgraph Isomorphism	**no**	**no**	**no**
Clique	**no**	**no**	**no**

Table 1. The problems considered in this paper. If a problem falls under one of the frameworks, that is, satisfies the conditions of the corresponding meta-classification, we indicate this with "yes", and if not, with "no" (in bold). If this is unknown, then we indicates this with a "?". The list of problems is not exhaustive. The problems were chosen to illustrate the wide reach of the frameworks and their differences. For many of the problems, the results follow from the literature or by combining known results in the literature. This is explained in detail in Sections 4 and 5.

Both problems are polynomial-time solvable for subcubic graphs, as shown by Ueno, Kajitani and Gotoh [75]. Munaro [66] showed that even Weighted Feedback Vertex Set is polynomial-time solvable for subcubic graphs. The same result also holds for Independent Feedback Vertex Set [52]. However, Connected Vertex Cover and (Independent) Feedback Vertex Set satisfy C1 [7] and are NP-complete for planar graphs of maximum degree at most 4, as shown in [38] for Connected Vertex Cover and in [73] for Feedback Vertex Set (and by taking 1-subdivisions the same holds for Independent Feedback Vertex Set). Hence, these three problems do satisfy the conditions of Theorem 1.

We also know of problems that satisfy the conditions of Theorem 2 (and thus of Theorem 1) but not those of Theorem 3. For example, Hamilton Cycle is solvable in polynomial-time for graphs of bounded treewidth [8], so satisfies C1, and it is NP-complete for planar subcubic graphs [40] (even if they are also bipartite and have arbitrarily large girth [66]). Hence, Hamilton Cycle satisfies the conditions of Theorem 2, and also satisfies C2. However, unlike its generalization Long Cycle, which is C123, Hamilton Cycle does not satisfy C3 [61], so it is not a C123-problem. The same holds for Hamilton Path (which contrasts the C123-property of Long Path).

Finally, there also exist problems that satisfy the conditions of Theorem 3, and thus are C123, but that do not satisfy the conditions of Theorems 1 and 2. Namely, Max-Cut is polynomial-time solvable for planar graphs [46] (and thus also for planar subcubic graphs). However, we show in Section 4 that Max-Cut satisfies the conditions of Theorem 3, that is, is a C123-problem.

Apart from Max-Cut and possibly Tree-Width and Perfect Matching Cut, all C123-problems from Section 4 are NP-complete for planar subcubic graphs, and in the proofs of Section 4 we will make explicit observations about this. So they satisfy the conditions of Theorem 2 (and thus of Theorem 1). The complexity of Tree-Width and Perfect Matching Cut is still open for planar graphs and planar subcubic graphs. It is also still open whether Diameter and Radius allow a distinction between almost-linear-time solvability versus not having a subquadratic-time algorithm on planar and subcubic planar graphs.

2 The Proof of Theorem 3 and a Potential Generalization

In this section we prove Theorem 3 by showing a potentially stronger result. We also show a similar strengthening of Corollary 3. For doing this, we must first define some terminology, which we will also need for other parts of our paper.

A *tree decomposition* of a graph $G = (V, E)$ is a pair $(T, \mathcal{X})$ where T is a tree and $\mathcal{X}$ is a collection of subsets of V called *bags* such that the following holds. A vertex $i \in T$ is a *node* and corresponds to exactly one bag $X_i \in \mathcal{X}$. The tree T has the following two properties. First, for each $v \in V$, the nodes of T that contain v induce a non-empty connected subgraph of T. Second, for each edge $vw \in E$, there is at least one node of T that contains both v and w. The *width* of $(T, \mathcal{X})$ is one less than the size of the largest bag in $\mathcal{X}$. The *treewidth* of G is the minimum width of its tree decompositions. If we require T to be a path, then we obtain the notions *path decomposition* and *path-width*.

A graph parameter p *dominates* a parameter q if there is a function f such that $p(G) \leq f(q(G))$ for every graph G. If p dominates q but q does not dominate p, then p is *more powerful* than q. If p dominates q and vice versa, then we say that p and q are *equivalent*. Note that every graph of path-width at most c has treewidth at most c. However, the class of trees has treewidth 1, but unbounded path-width (see [31]). Hence, treewidth is more powerful than path-width.

In order to show our potentially stronger result (Theorem 5) we replace C1 by the new condition C1′, and we call a graph problem that satisfies C1′, C2 and C3, a *C1′23-problem*.

C1′. *Π* is efficiently solvable for every graph class of bounded path-width;

We also need the following well-known theorem from Bienstock, Robertson, Seymour and Thomas.

Theorem 4 ([9]). *For every forest F, all F-minor-free graphs have path-width at most $|V(F)| - 2$.*

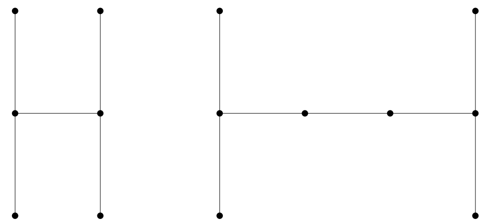

Fig. 2. Left: the graph H_1. Right: the graph H_3.

To prove our result, we also make use of an easy observation, which is well-known (see, e.g., [5]). Let H_1 be the "H"-graph, which is formed by an edge (the *middle edge*) joining the middle vertices of two paths on three vertices. For $\ell \geq 2$, let H_ℓ be the graph obtained from H_1 by subdividing the middle edge exactly $\ell - 1$ times. See Fig. 2 for two examples. Now, every problem that satisfies C2 and C3 is NP-hard for the class of $(C_3, \ldots, C_\ell, H_1, \ldots, H_k)$-subgraph-free graphs for every pair of integers k, ℓ.

We can now give the proof of our result.

Theorem 5. *Let Π be a C1′23-problem. For any finite set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-subgraph-free graphs is efficiently solvable if $\mathcal{H}$ contains a graph from $\mathcal{S}$ and computationally hard otherwise.*

Proof. For $p \geq 1$, let $\mathcal{H} = \{H_1, \ldots, H_p\}$. First assume $\mathcal{H}$ has a graph, say H_1, from $\mathcal{S}$. As G is $\mathcal{H}$-subgraph-free, G is H_1-subgraph-free. It is well known (see e.g. [43,44]) that a H-subgraph-free graph with $H \in \mathcal{S}$ is H-minor-free. Hence, G is H_1-minor-free so, by Theorem 4, G has constant path-width at most $|V(H_1)| - 2$, so we can solve Π efficiently by C1′.

Now assume that $\mathcal{H}$ contains no graph from $\mathcal{S}$. Let ℓ_1 be the size of a largest cycle of the graphs in $\mathcal{H}$; set $\ell_1 := 1$ if these graphs are all trees. Let ℓ_2 be the largest distance between a pair of vertices that are each of degree at least 3 and that both belong to the same connected component of a graph in $\mathcal{H}$; set $\ell_2 := 1$ if no such pair exists. Let $\ell = \max\{\ell_1, \ell_2\}$; note that $\ell \geq 1$.

Let $\mathcal{G}$ be the class of subcubic graphs. By C2, Π is hard for $\mathcal{G}$. By C3, there is a $k \geq 1$ such that Π is also hard for $\mathcal{G}^{kp}$ for every integer p. Set $p := \ell$. It remains to show that every graph in $\mathcal{G}^{k\ell}$ is $\mathcal{H}$-subgraph-free. Let $H \in \mathcal{H}$. As $H \notin \mathcal{S}$, H has either a vertex of degree at least 4, a cycle or a connected component with two vertices of degree 3. If H has a vertex of degree at least 4, that is, contains $K_{1,4}$ as a subgraph, then G is H-subgraph-free, as G is subcubic and thus $K_{1,4}$-subgraph-free. If H has a cycle, then G is H-subgraph-free, as the cycle in H has length at most $\ell_1 \leq \ell$. If H has a connected component with two vertices x and y of degree 3, then G is H-subgraph-free, as the path between x and y in H has length at most $\ell_2 \leq \ell$. □

Theorem 3 now follows from Theorem 5, as every graph class of bounded path-width has bounded treewidth.

Even though treewidth is more powerful than path-width, this may change if we restrict ourselves to some special graph class. For example, Hickingbotham [48] recently proved that for every finite set of graphs $\mathcal{H}$, the class of $\mathcal{H}$-free graphs has bounded treewidth if and only if it has bounded path-width (see also Section 6.4). Similarly, for every finite set of graphs $\mathcal{H}$, the class of $\mathcal{H}$-subgraph-free graphs has bounded treewidth if and only if it has bounded path-width.[4] Hence, we can strengthen Corollary 3 as follows.

Corollary 4. *Let Π be a C1′23-problem. For any finite set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-subgraph-free graphs is efficiently solvable if the class of $\mathcal{H}$-subgraph-free graphs has bounded path-width and computationally hard otherwise.*

In general, if some width parameter p is less powerful than some other width parameter q, then more problems are efficiently solvable for graph classes with bounded p than for graph classes with bounded q. However, so far, we are not aware of any graph problem that satisfies C1′, C2 and C3, but not the original condition C1. Hence, we have *no* evidence yet if Theorem 5/Corollary 4 is stronger than Theorem 3/Corollary 3.

[4] If $\mathcal{H}$ has a graph from $\mathcal{S}$, repeat the arguments from the proof of Theorem 5 to find that the class of $\mathcal{H}$-subgraph-free graphs has bounded path-width. Else, the class of $\mathcal{H}$-free graphs has unbounded tree-width, as mentioned before.

3 Limitations of the Framework

In this section, we discuss some limitations of our framework.

3.1 Forbidding An Infinite Number of Subgraphs

Unlike Theorems 1 and 2, the set of graphs $\mathcal{H}$ in Theorems 3 and 5 cannot be allowed to have infinite size. This is because there exist infinite sets $\mathcal{H}$ such that

1. $\mathcal{H}$ contains no graphs from $\mathcal{S}$.
2. All C123-problems are efficiently solvable on $\mathcal{H}$-subgraph-free graphs.

To illustrate this, we give two examples. See, e.g. [53], for another example.

Example 1. Let $\mathcal{H}$ be the set of cycles $\mathcal{C}$. No graph from $\mathcal{C}$ belongs to $\mathcal{S}$. Every $\mathcal{C}$-subgraph-free graph is a forest and thus has treewidth 1. Hence, every C123-problem is efficiently solvable on $\mathcal{C}$-subgraph-free graphs (as it satisfies condition C1).

Example 2. Let $\mathcal{H} = \{H_1, H_2, \ldots\}$; see also Fig. 2. No graph from $\mathcal{H}$ belongs to $\mathcal{S}$. Every $\mathcal{H}$-subgraph-free graph G is H_1-minor-free. By Theorem 4, G has path-width, and thus treewidth, at most 4. Hence, every C123-problem is efficiently solvable on $\mathcal{H}$-subgraph-free graphs.

3.2 Relaxing Condition C3

In C3 we require the class $\mathcal{G}$ to be subcubic. In this way we are able to construct a class $\mathcal{G}^{k\ell}$ in the proof of Theorem 5 that is $(K_{1,4}, C_3, \ldots, C_\ell, H_1, \ldots, H_\ell)$- subgraph-free.[5] If we allow $\mathcal{G}$ to be any graph class instead of requiring $\mathcal{G}$ to be subcubic, then the proof of Theorem 5 no longer works. That is, following the same arguments we can only construct a graph class that due to C2, is either $K_{1,4}$-subgraph-free (or equivalently, subcubic) or, due to C3, is $(C_3, \ldots, C_\ell, H_1, \ldots, H_\ell)$-subgraph-free. Consequently, in that case, we can only obtain a dichotomy for $\mathcal{H}$-subgraph-free graphs if $|\mathcal{H}| = 1$. This relaxation could potentially lead to a classification of more problems. However, so far, we have not identified any problems that belong to this relaxation but not to our original framework.

We also note that the existence of a problem-specific integer $k \geq 1$ in C3 is needed. For instance, a 1-division is bipartite and some computationally hard problems, such as Independent Set, become efficiently solvable on bipartite graphs. In the proofs in Section 4, k takes on values 1, 2, 3 and 4.

3.3 Not Being C123 But Having the Same Classification

We give a general construction for problems that are not C123 but that still have the same classification as in Theorem 3. This implies that the C123 condition alone is not sufficient to explain why problems can be classified in this way. We start by defining an infinite graph class $\mathcal{B}$ with the following four properties:

1. For every integer p, the p-subdivision of any graph in $\mathcal{B}$ is not in $\mathcal{B}$.
2. We can recognize whether a graph belongs to $\mathcal{B}$ in polynomial time.
3. Every graph in $\mathcal{B}$ admits a 3-colouring.
4. For every finite set $\mathcal{H}$ disjoint from $\mathcal{S}$, there is an $\mathcal{H}$-subgraph-free graph in $\mathcal{B}$.

[5] The problems in [37,66] are shown to be NP-complete for planar subcubic graphs of high girth, whereas we consider subcubic graphs of high girth that do not contain any small subdivided "H"-graph as a subgraph (the *girth* of a graph G that is not a forest is the length of a shortest cycle in G).

As an example of $\mathcal{B}$, take the set of all graphs obtained from a cycle after adding a new vertex made adjacent to precisely one vertex of the cycle.

The problem $\mathcal{B}$-Modified List Colouring takes as input a graph G with a list assignment L. The question is if G has both a connected component that is a graph from $\mathcal{B}$ and a colouring respecting L. As List Colouring is NP-complete for subcubic graphs, $\mathcal{B}$-Modified List Colouring is NP-complete for subcubic graphs. However, due to Property 1, $\mathcal{B}$-Modified List Colouring does not satisfy C3. Nevertheless we still obtain the same complexity classification.

Theorem 6. *For any finite set of graphs $\mathcal{H}$, $\mathcal{B}$-Modified List Colouring on $\mathcal{H}$-subgraph-free graphs is polynomial-time solvable if $\mathcal{H}$ contains a graph from $\mathcal{S}$ and NP-complete otherwise.*

Proof. First suppose $\mathcal{H}$ has a graph from $\mathcal{S}$. As List Colouring is C123, it is polynomial-time solvable on $\mathcal{H}$-subgraph-free graphs. By Property 2, we can check in polynomial time if a graph has a connected component in $\mathcal{B}$. Hence, $\mathcal{B}$-Modified List Colouring is polynomial solvable on $\mathcal{H}$-subgraph-free graphs.

Now suppose $\mathcal{H}$ has no graph from $\mathcal{S}$. Then List Colouring, being C123, is NP-complete for $\mathcal{H}$-subgraph-free graphs. Let (G, L) be an instance of List Colouring where G is $\mathcal{H}$-subgraph-free. By Property 4, there is an $\mathcal{H}$-subgraph-free graph $B \in \mathcal{B}$. Let $G' = G + B$. Extend L to a list assignment L' by giving each vertex of B list $\{1, 2, 3\}$. We claim that (G, L) is a yes-instance of List Colouring if and only if (G', L') is a yes-instance of $\mathcal{B}$-Modified List Colouring.

First suppose G has a colouring respecting L. By Property 3, B is 3-colourable. As vertices of B have list $\{1, 2, 3\}$, G' has a colouring respecting L'. As G has $B \in \mathcal{B}$ as a connected component, (G', L') is a yes-instance of $\mathcal{B}$-Modified List Colouring. Now suppose that (G', L') is a yes-instance of $\mathcal{B}$-Modified List Colouring. Then, G' has a colouring respecting L', and thus G has a colouring respecting L. ◻

4 NP-Complete Problems

In Section 4.1, we give examples of width parameter problems that are C123. In Section 4.2 we give examples of partitioning, covering and packing problems that are C123. In Section 4.3 we show the same for a number of network design problems. In fact we do a bit more. Namely, we also show that these problems belong to the minor and topological minor frameworks whenever this is indicated in Table 1. This comes down to showing NP-completeness for subcubic planar graphs. We do this either by citing a known result from the literature or by giving an explicit proof. We will not explicitly remark this in the remainder of the section.

4.1 Width Parameter Problems

Let Path-Width and Tree-Width be the problems of deciding for a given integer k and graph G, if G has path-width, or respectively, treewidth at most k.

Theorem 7. Path-Width *is a C123-problem.*

Proof. Path-Width is linear-time solvable for every graph class of bounded treewidth [17] so satisfies C1. It is NP-complete for 2-subdivisions of planar cubic graphs [65] so satisfies C2. It also satisfies C3, as we will prove the following claim:

Claim. A graph $G = (V, E)$ that is a 2-subdivision of a graph G'' has path-width k if and only if the 1-subdivision G' of G has path-width k.

To prove the claim, we use the equivalence of path-width to the vertex separation number [55]. We recall the definition. Let L be a bijection between V and $\{1, \ldots, |V|\}$, also called a *layout* of G. Let $V_L(i) = \{u \mid L(u) \leq i \text{ and } \exists v \in V : uv \in E \text{ and } L(v) > i\}$. Then $\text{vs}_L(G) = \max_{i \in \{1,\ldots,|V|\}}\{|V_L(i)|\}$ and $\text{vs}(G) = \min_L\{\text{vs}_L(G)\}$.

Following Kinnersley [55], G has a layout L such that $\text{vs}_L(G) = k$. In a 2-subdivision, any edge uv gets replaced by edges ua, ab, and bv, where a and b are new vertices specific to the edge uv. In a *standard layout* L for G, $L(a) = L(b) - 1$ and $L(u) < L(a)$. Following Ellis, Sudborough and Turner [32], we may assume that L is a standard layout.

For some edge uv of G'', consider a further subdivision of each of ua, ab, and bv. Let x, y, z be the newly created vertices respectively. Modify L by placing x directly before a, y between a and b, and z directly after b. Let L' denote the new layout. For simplicity and abusing notation, we use $L'(x) = L(a) - \frac{1}{2}$, $L'(y) = L(a) + \frac{1}{2} = L(b) - \frac{1}{2}$ and $L'(z) = L(b) + \frac{1}{2}$ to denote the positions of x, y and z in the new layout respectively. For any $i < L(a) - \frac{1}{2}$, $V_{L'}(i) = V_L(i)$, because $L(a) > i$ and $L'(x) > i$. Next, we observe that $V_{L'}(L'(x)) = V_{L'}(L(a) - \frac{1}{2}) = (V_L(L(a)) \setminus \{a\}) \cup \{x\}$, because b follows after u in L and now a follows after x in L'. Hence, it has the same size as $V_L(L(a))$, at most k. Similarly, we can observe that $V_{L'}(L(a)) = V_L(L(a))$ (note that $L'(u) = L(u) < L(a)$), $V_{L'}(L(a) + \frac{1}{2}) = (V_L(L(a)) \setminus \{a\}) \cup \{y\}$, and $V_{L'}(L(b)) = (V_L(L(a)) \setminus \{a\}) \cup \{b\}$, which all have size at most k. We then observe that $V_{L'}(L(b) + \frac{1}{2})$ is equal to $V_L(b)$ with b replaced by z if $b \in V_L(b)$. Similarly, for any $i > L(b)$, if $b \in V_L(i)$, we can replace it by z to obtain $V'_L(i)$; otherwise, $V_{L'}(i) = V_L(i)$. Note that a is never part of $V_L(i)$. In all cases, the size remains bounded by k. Hence, $\mathrm{vs}_{L'} \leq k$ and by the aforementioned equivalence between path-width and vertex separation number [55], G' has pathwidth at most k.

As subdivision cannot decrease path-width (or considering the converse, contraction cannot increase it), the claim is proven. This finishes the proof of Theorem 7. ◻

Theorem 8. Tree-Width *is a C123-problem.*

Proof. Tree-Width is linear-time solvable for every graph class of bounded treewidth [11]. Very recently, it was announced that Tree-Width is NP-complete for cubic graphs [12] so the problem satisfies C2. It also satisfies C3, as we will prove the following claim:

Claim. A graph G has treewidth k if and only if the 1-subdivision of G has treewidth k.

Taking a minor of a graph does not increase its treewidth so the treewidth cannot decrease after subdividing an edge. If a graph G has treewidth 1, then G remains a tree after subdividing an edge. Suppose G has treewidth at least 2. Let $(T, \mathcal{X})$ be a tree decomposition of G with width $k \geq 2$. Let G' be the graph obtained by replacing an edge uv with edges ux and xv for a new vertex x. Pick an arbitrary bag B from $\mathcal{X}$ containing u and v. Introduce the bag $\{u, v, x\}$ and make the corresponding node adjacent to the B-node. This yields a tree decomposition of G' of width k, as $k \geq 2$ and the bag we added has size 3. Hence, the claim and thus the theorem is proven. ◻

4.2 Partitioning, Covering and Packing Problems

The Vertex Cover problem is to decide if a graph has a vertex cover of size at most k for some given integer k. The Independent Set problem is to decide if a graph has an independent set of size at least k for some given integer k. Note that Vertex Cover and Independent Set are polynomially equivalent. In the following theorem we recover the classification of [6].

Theorem 9. Vertex Cover *and* Independent Set *are C123-problems.*

Proof. Both are linear-time solvable for graphs of bounded treewidth [8] so satisfy C1. Both are NP-complete for 2-connected cubic planar graphs [64] so satisfy C2. They also satisfy C3, as a graph G on m edges has an independent set of size k if and only if the 2-subdivision of G has an independent set of size $k + m$ [68]. ◻

A set $D \subseteq V$ is *dominating* a graph $G = (V, E)$ if every vertex The (Independent) Dominating Set problem is to decide if a graph has an (independent) dominating set of size at most k for some integer k. A set $F \subseteq E$ is an *edge dominating set* if every edge in $E \setminus F$ shares an end-vertex with an edge of F. The corresponding decision problem is Edge Dominating Set. Recall that this problem is polynomially equivalent to Minimum Maximal Matching [47].

The following theorem shows that both problems are C123, just like Dominating Set; hence, we recover the classification of [6] for the latter problem.

Theorem 10. Dominating Set, Independent Dominating Set *and* Edge Dominating Set *are C123-problems.*

Proof. Dominating Set [8], Independent Dominating Set [74] and Edge Dominating Set [7] are linear-time solvable for graphs of bounded treewidth so satisfy C1. Dominating Set [41], Independent Dominating Set [23] and Edge Dominating Set [82] are NP-complete for planar subcubic graphs so satisfy C2. For showing C3 we use the following claim (see for example [22]) for a proof). A graph G with m edges has a dominating set, independent dominating set or edge dominating set of size k if and only if the 3-subdivision of G has a dominating set, independent dominating set or edge dominating set, respectively, of size $k+m$. ☐

For a graph $G=(V,E)$, a function $c : V \to \{1,2\ldots\}$ is a *colouring* of G if $c(u) \neq c(v)$ for every pair of adjacent vertices u and v. If $c(V)=\{1,\ldots,k\}$ for some integer $k \geq 1$, then c is also said to be a *k-colouring.* Note that a k-colouring of G partitions V into k independent sets, which are called *colour classes.* The 3-Colouring problem is to decide if a graph has a 3-colouring. A *list assignment* of a graph $G=(V,E)$ is a function L that associates a *list of admissible colours* $L(u) \subseteq \{1,2,\ldots\}$ to each vertex $u \in V$. A colouring c of G *respects* L if $c(u) \in L(u)$ for every $u \in V$. The List Colouring problem is to decide if a graph G with a list assignment L has a colouring that respects L. An *odd cycle transversal* in a graph $G=(V,E)$ is a subset $S \subseteq V$ such that $G-S$ is bipartite. If S is independent, then S is an *independent odd cycle transversal.* The (Independent) Odd Cycle Transversal problem is to decide if a graph has an (independent) odd cycle transversal of size at most k for a given integer k. Note that a graph has an independent odd cycle transversal of size at most k if and only if it has 3-colouring in which one of the colour classes has size at most k.

Recall that 3-Colouring is not a C123-problem, as it does not satisfy C2 (see also Table 1). This is in contrast to the situation for List Colouring, Odd Cycle Transversal and Independent Odd Cycle Transversal: we show that all three problems are C123, and in this way recover the classification of [43] for List Colouring. Our proof shows in particular that all three problems are in fact NP-complete for planar subcubic graphs.

Theorem 11. List Colouring*,* Odd Cycle Transversal*, and* Independent Odd Cycle Transversal *are C123-problems.*

Proof. Jansen and Scheffler [50] proved that List Colouring can be solved in linear time on graphs of bounded treewidth, so satisfies C1. Both Odd Cycle Transversal and Independent Odd Cycle Transversal are linear-time solvable for graphs bounded treewidth [7,35] so satisfy C1.

To prove C2 for all three problems, we modify the standard reduction to 3-Colouring for planar graphs, which is from Planar 3-Satisfiability (we use the reduction form Proposition 2.27 of [42]). This enables us to prove that all three problems are NP-complete even for planar subcubic graphs.

The problem Planar 3-Satisfiability is known to be NP-complete even when each literal appears in at most two clauses (see Theorem 2 in [27]). It is defined as follows. Given a CNF formula ϕ that consists of a set $X=\{x_1,x_2,...,x_n\}$ of Boolean variables, and a set $C=\{C_1,C_2,...,C_m\}$ of two-literal or three-literal clauses over X, does there exist a truth assignment for X such that each C_j contains at least one true literal? If such a truth assignment exists, then ϕ is *satisfiable.*

Let ϕ be an instance of Planar 3-Satisfiability on n variables and m clauses. From ϕ we construct a graph G as follows:

- For $i=1,\ldots,n$, add the *literal vertices* x_i and $\overline{x_i}$ and the edge $x_i\overline{x_i}$.
- Add a path P of $2m$ vertices. The odd vertices represent *false* and the even vertices *true.*
- For each clause C_j, add a *clause gadget* as in Fig. 3 with three labelled vertices $c_{j_1}, c_{j_2}, c_{j_3}$ as well as an *output* vertex labelled c_j.
- Fix an order of the literals $x_{j_1}, x_{j_2}, x_{j_3}$ of each three-literal clause C_j and for $h=1,\ldots,3$, identify x_{j_h} with c_{j_h}.
- Fix an order of the literals x_{j_1}, x_{j_2} of each two-literal clause C_j and for $h=1,\ldots,2$, identify x_{j_h} with c_{j_h}.
- Add an edge between c_i and the ith odd vertex (representing false) of P.
- Add an edge between any unused input c_{i_3} and the ith even vertex (representing even) on P.

Note that G is subcubic. Let us argue that G is also planar. In Planar 3-Satisfiability, the bipartite incidence graph of clauses with variables is planar. We build G so as to be planar in the following way. Uppermost we place the literals assigned to the inputs of the clause gadgets in just the manner prescribed in the bipartite incidence graph. Lowermost, we place the path of length $2m$ that will be joined on the odd vertices to the output nodes of the clause gadgets.

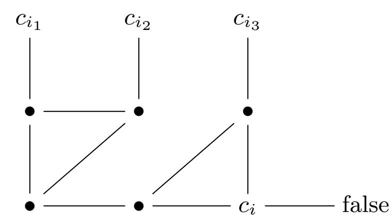

Fig. 3. The clause gadget in the reduction from (Planar) 3-Satisfiability to 3-Colouring drawn with an edge connecting the output node to a vertex representing false. The property that the gadget enforces is that not all of the three input nodes c_{i_1}, c_{i_2}, c_{i_3} may be coloured the same as the vertex representing false.

We will first prove that G has an independent odd cycle transversal of size $2m$ if and only if ϕ is satisfiable. First suppose that G has an independent odd cycle transversal $S = \{v_1, \ldots, v_{2m}\}$ of size $2m$. As G contains $2m$ triangles, two in each clause gadget, $T_1, \ldots, T_{2m}$, we may assume without loss of generality that $v_i \in V(T_i)$ for every $i \in \{1, \ldots, 2m\}$. Note that $G - S$ is bipartite by the definition of an odd cycle transversal. Thus we can find a 3-colouring of G by colouring every vertex of S with colour 1 and colouring every vertex of $G-S$ with colours 2 and 3. As each literal vertex belongs to $G-S$, it is assigned either colour 2 or colour 3, just like each vertex of P. Let us assume, without loss of generality, that the odd vertices on this path are coloured 3. Hence, 2 represents true and 3 false. But now, by construction of the clause gadget, at least one of the vertices c_{j_1}, c_{j_2} and c_{j_3} is coloured 2 and is identified with a literal for $j = 1, \ldots m$, and therefore we deduce that ϕ is satisfiable.

Now suppose that ϕ is satisfiable. Colour the vertices of P with colours alternating between 3 and 2. In each clause, colour each true literal with colour 2 and each false or unused literal with colour 3. Then, by construction of the clause gadget we can extend this to a 3-colouring of G. Let S be the set of vertices of G coloured 1. Then $S \subseteq V(T_1) \cup \cdots \cup V(T_{2m})$, and thus S consists of exactly one vertex of each T_i. So S is an independent odd cycle transversal of G of size $2m$.

To prove C2 for List Colouring, we use exactly the reduction above, with the literal vertices, any unused input c_{i_3} and the vertices of P assigned the list $\{2, 3\}$, but all other vertices permitted to be any of the three colours. Hence, as every list will be a subset of $\{1, 2, 3\}$, this result even holds for List 3-Colouring.

We now prove C2 for Independent Odd Cycle Transversal and C3 for Odd Cycle Transversal and Independent Odd Cycle Transversal. To prove C3 for Odd Cycle Transversal, we can just use the following claim (see for example [21] for a proof):

Claim. The size of a minimum odd cycle transversal of G is equal to the size of a minimum odd cycle transversal of the 2-subdivision of G.

We now show C2 and C3 for Independent Odd Cycle Transversal by proving that Independent Odd Cycle Transversal is NP-complete for $2p$-subdivisions of subcubic planar graphs. Consider the subclass of planar subcubic graphs that correspond to instances of Planar 3-Satisfiability as defined in our proof for C2 for Odd Cycle Transversal. We now apply the above Claim sufficiently many times. In the graph G' resulting from the $2p$-subdivision, any minimum odd cycle transversal will also be an independent odd cycle transversal (by inspection of the proof for C2 for Odd Cycle Transversal, because the cycles that were once triangles become further and further apart).

By inspection of the proof of Lemma 3 in [43], also List Colouring satisfies C3. ◻

A P_3*-factor* or *perfect* P_3*-packing* of a graph $G = (V, E)$ with $|V| = 3k$ for some integer $k \geq 1$ is a partition of V into subsets $V_1, \ldots, V_k$, such that each $G[V_i]$ is either isomorphic to P_3 or K_3. The corresponding decision problem, which asks whether a graph has such a partition, is known as P_3-Factor or Perfect P_3-Packing. We show that P_3-Factor is a C123-problem, a result which is essentially due to [5].

Theorem 12. P_3-Factor *is a C123-problem.*

Proof. This follows from combining Proposition 1 of [5] for showing C1 with Lemma 12 of [5] for showing C2 and C3 (with $k = 3$). Recently, Xi and Lin [80] proved that P_3-Factor is NP-complete even for claw-free planar cubic graphs, which also proves C2. ◻

4.3 Network Design Problems

A *(vertex) cut* of a graph $G = (V, E)$ is a partition $(S, V \setminus S)$ of V. The *size* of $(S, V \setminus S)$ is the number of edges with one end in S and the other in $V \setminus S$. The Max-Cut problem is to decide if a graph has a cut of size at least k for some integer k. By combining the next result with Theorem 3, we recover the classification of [53].

Theorem 13 ([53]). Max-Cut *is a C123-problem.*

Proof. Max-Cut is linear-time solvable for graphs of bounded treewidth [7] and NP-complete for subcubic graphs [81] so satisfies C1 and C2. A cut C of a graph G is *maximum* if G has no cut of greater size. Kamiński [53] proved that a graph $G = (V, E)$ has a maximum cut of size at least c if and only if the 2-subdivision of G has a maximum cut of size at least $c + 2|E|$. This shows C3. ◻

Let $G = (V, E)$ be a graph. A set $M \subseteq E$ is a *perfect matching* if no two edges in M share an end-vertex and moreover, every vertex of the graph is incident to an edge of M. A set $M \subseteq E$ is an *edge cut* of G if it is possible to partition V into two sets B and R, such that M consists of all the edges with one end-vertex in B and the other one in R. A set $M \subseteq E$ is a *perfect matching cut* of G if M is a perfect matching that is also an edge cut. The Perfect Matching Cut is to decide if a graph has a perfect matching cut. Lucke et al. [34] recently showed that Perfect Matching Cut is C123.

Theorem 14 ([34]). Perfect Matching Cut *is a C123-problem.*

Proof. Le and Telle [59] observed that Perfect Matching Cut is polynomial-time solvable for graphs of bounded treewidth, and that for every integer $g \geq 3$, it is NP-complete even for subcubic bipartite graphs of girth at least g. Hence, Perfect Matching Cut satisfies C1 and C2. Lucke et al. [34] showed the problem is C123 by proving C3, using $k = 4$.

Given a graph G and a set of terminals $T \subseteq V(G)$, and an integer k, the problems Edge (Node) Steiner Tree are to decide if G has a subtree containing all the terminals of T, such that the subtree has at most k edges (vertices).

Theorem 15. Edge *and* Node Steiner Tree *are C123-problems.*

Proof. As the two variants are equivalent (on unweighted graphs), we only consider Edge Steiner Tree, which is linear-time solvable for graphs of bounded treewidth [7] so satisfies C1. For showing C2, we reduce from Edge Steiner Tree, which is NP-complete even for grid graphs [38], and thus for planar graphs.

Let (G, T, k) be an instance, where G is a planar graph with $|V(G)| = n$. We build a planar subcubic graph G' where we replace each node v in G with a rooted binary tree T_v in which there are n leaf vertices (so the tree contains at most $2n$ nodes and is of depth $\lceil \log n \rceil$). For each edge $e = uv$ of G, add to G' a path e' of length $4n^2$ between some a leaf of T_u and a leaf of T_v (ensuring that each leaf is incident with at most one such path). If v in G is in T, then the root vertex of T_v is a terminal in G' (and these are the only terminals in G' and form the set T'). We note that G is planar subcubic, and we claim that (G,T,k) is a yes-instance if and only if $(G',T',4n^2 \cdot k + 2n^2)$ is a yes-instance.

First suppose G has a Steiner tree S with at most k edges. We build a Steiner tree S' in G': if $e = uv$ is in S, then we add to S' a path that comprises e' and paths that join the roots of T_u and T_v to e'. The sum of the lengths of these paths, additional to the $4n^2 \cdot k$, is bounded above by $2 \cdot n \cdot \log n \leq 2n^2$.

Now suppose G' has a Steiner tree S' with at most $4n^2 \cdot k + 2n^2$ edges. We build a tree S in G: if $e = uv$ and e' is in S', we add e to S. Then S is a Steiner tree in G. As the length of a path from T_u to T_v is $4n^2$, the sum of the lengths of all such paths in S' is a whole multiple of $4n^2$, so $|E(S)| \leq k$.

To prove C3, it suffices to show the following claim:

Claim. A graph G has an edge Steiner tree for terminals T of size at most k if and only if the 1-subdivision of G has an edge Steiner tree for terminals T of size at most $2k$.

In order to see this, let G' be the 1-subdivision of G. Let e_1 and e_2 be the two edges obtained from subdividing an edge $e \in E(G)$. Given a Steiner tree S of G with at most k edges, we obtain a Steiner tree of G' with at most $2k$ edges by replacing each edge e of S with e_1 and e_2. Given a Steiner tree S' of G' with at most $2k$ edges, we may assume that for any edge e of G, either neither or both of e_1 and e_2 are in S'; if S' contains only one it can safely be discarded. To obtain a Steiner tree of G with at most k edges, include each edge e if both e_1 and e_2 are in S'. ◻

For a graph $G = (V,E)$, set of terminals $T \subseteq V$ and integer k, the Edge (Node) Multiway Cut problem is to decide if there is a set $S \subseteq E$ ($S \subseteq V \setminus T$) such that $|S| \leq k$ and for every pair $\{u,v\} \in T$, $G \setminus S$ has no path between u and v.

Theorem 16. *Edge and Node Multiway Cut are C123-problems.*

Proof. Edge Multiway Cut is linear-time solvable for graphs of bounded treewidth [30] (also following [7]) and NP-complete for planar subcubic graphs [51] so satisfies C1 and C2. It satisfies C3 as well, as we will prove the following claim:

Claim. A graph G has an edge multiway cut for a set of terminals T of size at most k if and only if the 1-subdivision of G has an edge multiway cut for T of size at most k.

In order to see this, let G' be the 1-subdivision of G. For each edge e in G, there exist two edges in G'. If an edge of G is in an edge multiway cut for G and T, then it suffices to pick only one of the two edges created from it in G' to disconnect the paths e lies on. Vice versa, if an edge e' of G' is in an edge multiway cut for G' and T, then it suffices to pick the unique corresponding edge in G to disconnect the paths e' lies on.

We now turn to Node Multiway Cut, which is linear-time solvable for graph classes of bounded treewidth [7] (it is an extended monadic second-order linear extremum problem) and NP-complete for planar subcubic graphs [51] so satisfies C1 and C2. It satisfies C3, as we will prove the following claim:

A graph G has a node multiway cut for a set of terminals T of size at most k if and only if its 1-subdivision has a node multiway cut for T of size at most k.

In order to see this, let G' be the 1-subdivision of G. We observe that subdividing any edge of a graph does not create new connections between terminals. Moreover, we can assume that none of the newly introduced vertices of the subdivision are used in some optimal solution for G' and T. ◻

Given a graph G and disjoint vertex pairs $(s_1,t_1),(s_2,t_2),\dots(s_k,t_k)$, the Disjoint Paths problem is to decide if G has k pairwise vertex-disjoint paths from s_i to t_i for every i. We obtain the Induced Disjoint Paths problem if the paths are required to be mutually induced; a set of paths $P^1,\dots,P^k$ is *mutually induced* if $P^1,\dots,P^k$ are pairwise vertex-disjoint and there is no edge between a vertex of some P^i and a vertex of some P^j if $i \neq j$.

Theorem 17. *Disjoint Paths and Induced Disjoint Paths are C123-problems.*

Proof. The DISJOINT PATHS problem is linear-time solvable for graphs of bounded treewidth [71] and NP-complete for planar subcubic graphs [63] so satisfies C1 and C2. The INDUCED DISJOINT PATHS problem is solvable in polynomial time for graphs of bounded mim-width [49] and thus for bounded treewidth [76], so it satisfies C1. Let G' be the 1-subdivision of a subcubic graph G and let $\mathcal{T}$ be a set of disjoint vertex pairs. Then, $(G, \mathcal{T})$ is a yes-instance of DISJOINT PATHS if and only if $(G', \mathcal{T})$ is a yes-instance of DISJOINT PATHS if and only if $(G', \mathcal{T})$ is a yes-instance of INDUCED DISJOINT PATHS. Hence, C2 is satisfied for INDUCED DISJOINT PATHS as well and C3 is satisfied for both problems. ☐

The LONG PATH and LONG INDUCED PATH are to decide for a given graph G and integer k, whether G contains P_k as a subgraph or induced subgraph, respectively. The LONG CYCLE and LONG INDUCED CYCLE problems are defined similarly. By combining the next result with Theorem 3, we recover the classification of [53] for LONG PATH. The classification of LONG CYCLE was not made explicit in [5], but is implicitly there (combine Proposition 1 of [5] with Lemma 12 of [5]).

Theorem 18. *LONG PATH, LONG INDUCED PATH, LONG CYCLE and LONG INDUCED CYCLE are C123-problems.*

Proof. Bodlaender [10] proved that LONG PATH and LONG CYCLE are polynomial-time solvable for graphs of bounded treewidth. Hence, LONG PATH and LONG CYCLE satisfy C1. As HAMILTON PATH (so LONG PATH with $k = |V(G)|$) and HAMILTON CYCLE (so LONG CYCLE with $k = |V(G)|$) are NP-complete for subcubic planar graphs [39], LONG PATH and LONG CYCLE satisfy C2.

Let G' be the 1-subdivision of a subcubic graph G. Now the following holds: (G, k) is a yes-instance of LONG PATH if and only if $(G', 2k)$ is a yes-instance of LONG PATH if and only if $(G', 2k)$ is a yes-instance of LONG INDUCED PATH. Hence, C2 is satisfied for LONG INDUCED PATH as well, and C3 is satisfied for both problems. Moreover, LONG INDUCED PATH satisfies C1; it is even polynomial-time solvable for graphs of bounded mim-width [49]. We can make the same observations for LONG CYCLE and LONG INDUCED CYCLE. ☐

5 Polynomial-Time Solvable Problems

Let $d(s, t)$ denote the distance between s and t in a graph G. The *eccentricity* of $u \in V$ is $e(u) = \max_{v \in V} d(u, v)$. The *diameter* of G is the maximum eccentricity and the *radius* the minimum eccentricity. The DIAMETER and RADIUS problems are to find the diameter and radius, respectively, of a graph. We need a lemma.

Lemma 1. *Let G' be the 2-subdivision of a graph G with diameter d. Let d' be the diameter of G'. Then $3d \leq d' \leq 3d + 2$.*

Proof. Under edge-subdivision, the shortest path between two original vertices does not change, it is only of longer length. As the path between two adjacent vertices in G gets length 3 in G', use any diametral pair in G to find that $d' \geq 3d$.

Let u and v be two vertices of V'. If u and v belong to G, then they are of distance at most $3d$ in G'. If one of them, say u, belongs to V and the other one, v, belongs to $V' \setminus V$, then they are of distance at most $3d + 1$ in G', as any vertex in $V' \setminus V$ is one step away from some vertex in V and the diameter is d in G. If u and v both belong to $V' \setminus V$, then u is adjacent to some vertex $w_u \in V$ and v is adjacent to some vertex $w_v \in V$. As the diameter is d in G, vertices w_u and w_v lie at distance at most $3d$ from each other in G'. Hence, in this case, $d(u, v) \leq 3d + 2$ in G'. To summarize, the diameter of G' is at most $3d + 2$. ☐

Theorem 19. *Both DIAMETER and RADIUS are C123-problems.*

Proof. Both are solvable in $n^{1+o(1)}$ time for graphs of bounded treewidth [1] and thus satisfy Condition C1. Both also satisfy C2. Evald and Dahlgaard [33] proved that for subcubic graphs, no subquadratic algorithm exists for DIAMETER exists under the Orthogonal Vectors Conjecture [78], and no subquadratic algorithm exists for Radius exists under the Hitting Set Conjecture [1]. From the construction in the proof of the latter result, we observe that any constant subdivision of all edges of the graph does not affect the correctness of the reduction, i.e., the parameter p in the construction can be increased appropriately to account for the subdivisions of the other edges. Hence, Radius satisfies C3. By Lemma 1, Diameter satisfies C3 as well. □

6 Conclusions

By giving a meta-classification, we were able to unify a number of known results from the literature and give new complexity classifications for a variety of graph problems on classes of graphs characterized by a finite set $\mathcal{H}$ of forbidden subgraphs. Similar frameworks existed (even for infinite sets $\mathcal{H}$) already for the minor and topological minor relations, whereas for the subgraph relation, only some classifications for specific problems existed [6,43,53]. We showed that many problems belong to all three frameworks, and also that there exist problems that belong to one framework but not to (some of) the others.

In order to have stronger hardness results for our subgraph framework, we considered the unweighted versions of these problems. However, we note that most of the vertex-weighted and edge-weighted variants of these problems satisfy C1 as well; see [7]. We finish this section by setting out some directions for future work.

6.1 Refining and Extending the Subgraph Framework

We already defined in Section 2 the notions of one graph parameter p being more powerful than or equivalent with another graph parameter q. If neither p dominates q nor q dominates p, then p and q are *incomparable*. As mentioned, two non-equivalent parameters can become equivalent when we consider some special graph class. This could potentially lead to a more general framework (covering more graph problems).

Open Problem 1 *Determine if there exists a graph parameter p that is either less powerful than or incomparable to treewidth, such that*

(i) the parameters p and treewidth become equivalent for every class of $\mathcal{H}$-subgraph-free graphs, and
(ii) there exists a graph problem that is efficiently solvable for graph classes with bounded p, but that is computationally hard for graph classes of bounded treewidth.

In Section 2 we showed that path-width satisfies (i) but we do not know if path-width satisfies (ii). That is, we do not know any graph problem that satisfies C1′, C2 and C3, but not C1. We observe that an $n^{O(\mathrm{pw}^c)}$-time algorithm, where c is some constant, for solving some problem Π for graphs of bounded path-width leads to an $2^{O(\mathrm{tw}^c \log^{(c+1)} n)}$-time, so quasi-polynomial-time, algorithm for solving Π on graphs of bounded treewidth. This is due to the fact that the path-width of a graph G is at most $\log n$ times the treewidth of G [15]. Hence, Theorem 3 may also lead to a distinction between quasi-polynomial-time and being NP-complete (should such problems exist).

We describe three other approaches for refining or extending the subgraph framework. First, in Section 3.3, we gave a general construction for problems that are not C123, but that still have the same classification as in Theorem 3. This construction shows that there is some scope in refining the conditions C1, C2, C3. However, the construction is rather artificial. To increase our understanding of possible improvements of the conditions of our framework, addressing the following question might be helpful.

Open Problem 2 *Do there exist any natural graph problems that are not C123-problems, but that still have the same classification as in Theorem 3?*

To address Open Problem 2, we might need to extend the toolkit and consider other graph transformations than the edge subdivision in C3; see, e.g., [5] for examples of transformations that work for the induced subgraph relation.

As a second approach, we recall from Section 3.2 that we cannot relax condition C3 by allowing the class $\mathcal{G}$ to be an arbitrary graph class instead of being subcubic. If we do this nevertheless, we are only able to obtain a dichotomy for $\mathcal{H}$-subgraph-free graphs if $|\mathcal{H}| = 1$. This relaxation could potentially lead to a classification of more problems and we pose the following open problem.

Open Problem 3 *Can we classify more problems for H-subgraph-free graphs by no longer demanding that the class $\mathcal{G}$ in C3 is subcubic?*

So far, we have not identified any problems that belong to the relaxation but not to our original framework.

Recall that the set of forbidden graphs $\mathcal{H}$ is allowed to have infinite size in Theorems 1 and 2. For any infinite set of graphs $\mathcal{H}$, a C123-problem on $\mathcal{H}$-subgraph-free graphs is still efficiently solvable if $\mathcal{H}$ contains a graph H from $\mathcal{S}$. However, a C123-problem may no longer be computationally hard for $\mathcal{H}$-subgraph-free graphs if $\mathcal{H}$ has infinite size, as shown in Section 3.1 with some examples. Hence, as a third approach for extending the subgraph framework, we propose the following problem. This problem was also posed by Kamiński [53], namely for the C123-problem Max-Cut.

Open Problem 4 *Can we obtain dichotomies for C123-problems restricted to $\mathcal{H}$-subgraph-free graphs when $\mathcal{H}$ is allowed to have infinite size?*

In order to solve Open Problem 4, we need a better understanding of the treewidth of $\mathcal{H}$-subgraph-free graphs when $\mathcal{H}$ has infinite size. In recent years, such a study has been initiated for the induced subgraph relation; see, for example, [2,3,56,77] for many involved results in this direction.

6.2 Finding More Problems Falling under the Three Frameworks

There still exist many natural problems for which it is unknown whether they belong to the minor, topological minor or subgraph framework. For the first two frameworks, we recall the following open problems (note that in particular determining the complexity of these problems for planar graphs has been frequently stated as an open problem before).

Open Problem 5 *Determine the computational complexity of* Tree-Width *and* Perfect Matching Cut *for planar graphs and for planar subcubic graphs.*

Open Problem 6 *Determine the fine-grained complexity of* Diameter *and* Radius *for planar graphs and for planar subcubic graphs.*

We now turn to the subgraph framework. We showed that Tree-Width and Path-Width are C123, but further investigation might reveal more such problems that fit the subgraph framework.

Open Problem 7 *Do there exist other width parameters with the property that the problem of computing them is C123?*

We also made a full comparison between the minor, topological minor and subgraph frameworks (see Section 1.6). To increase our general understanding of the complexity of graph problems it would be interesting to find more problems that either belong to all frameworks or just to one or two.

6.3 Dropping One of the Conditions C1, C2, or C3

Another highly interesting direction is to investigate if we can obtain new complexity dichotomies for computationally hard graph problems that do not satisfy one of the conditions, C1, C2 or C3. We call such problems C23, C13, or C12, respectively. This direction leads to some interesting new research questions, and we are currently trying to determine new complexity classifications for a number of candidate problems; see, for example, [61] where C12-problems are considered. For all these problems, the complexity classifications will be different from the one in Theorem 3. In [61], this was shown for Hamilton Cycle, k-Disjoint Paths (for fixed $k \geq 2$) and C_k-Colouring (for odd k). To give another example, the NP-complete problem Clique is a C13-problem and is polynomial-time solvable for $\mathcal{H}$-subgraph-free graphs for every (possibly infinite) set of graphs $\mathcal{H}$, as we observed in Section 1.3.

However, we note that in general, obtaining complete classifications is challenging for C12-, C13- and C23-problems. In particular, we need a better understanding of the structure of P_r-subgraph-free graphs and H_i-subgraph-free graphs (recall that H_i is a subdivided "H"-graph). Recall that a graph is P_r-subgraph-free if and only if it is P_r-(topological)-minor-free. Hence, if a problem is open for the case where $H = P_r$ for one of the frameworks, then it is open for all three of them.

To illustrate the challenges with an example from the literature, consider the aforementioned Subgraph Isomorphism problem. This problem takes as input two graphs G_1 and G_2. Hence, it does not immediately fit in our framework, but one could view it as a C23-problem. The question is whether G_1 is a subgraph of G_2. Recall that the Subgraph Isomorphism problem is NP-complete even for input pairs (G_1, G_2) that are linear forests and thus even have path-width 1. Yet, even a classification for H-subgraph-free graphs is not straightforward; recall that Bodlaender et al. [16] settled the computational complexity of Subgraph Isomorphism for H-subgraph-free graphs but only for connected graphs H and leaving open the case where $H = P_5$. The latter case is open for the minor and topological minor frameworks as well due to the above observation.

6.4 The Induced Subgraph Relation

We finish our paper with some remarks on the induced subgraph relation. As mentioned, there exist ongoing and extensive studies on boundary graph classes (cf. [4,5,57,66]) and treewidth classifications (cf. [2,3,56,77]) in the literature. We note that for the induced subgraph relation, it is also useful to check C2 and C3. Namely, let Π be a problem satisfying C2 and C3. For any finite set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-free graphs is computationally hard if $\mathcal{H}$ contains no graph from $\mathcal{S}$. This follows from the same arguments as in the proof of Theorem 5.[6] Hence, if we aim to classify the computational complexity of problems satisfying C2 and C3 for H-free graphs (which include all C123-problems), then we may assume that $H \in \mathcal{S}$. For many of such problems, such as Independent Set, this already leads to challenging open cases.

As mentioned, we currently do not know even any algorithmic meta-theorem for the induced subgraph relation, not even for a single forbidden graph H. However, a recent result of Lozin and Razgon [60] provides at least an initial starting point. To explain their result, the *line graph* of a graph G has vertex set $E(G)$ and an edge between two vertices e_1 and e_2 if and only if e_1 and e_2 share an end-vertex in G. Let $\mathcal{T}$ be the class of line graphs of graphs of $\mathcal{S}$. Lozin and Razgon [60] showed that for any finite set of graphs $\mathcal{H}$, the class of $\mathcal{H}$-free graphs has bounded treewidth if and only if $\mathcal{H}$ contains a complete graph, a complete bipartite graph, a graph from $\mathcal{S}$ and a graph from $\mathcal{T}$. Their characterization leads to the following theorem, which could be viewed as a first meta-classification for the induced subgraph relation.

Theorem 20. *Let Π be a problem that is* NP*-complete on every graph class of unbounded treewidth, but polynomial-time solvable for every graph class of bounded treewidth. For every finite set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-free graphs is polynomial-time solvable if $\mathcal{H}$ contains a complete graph, a complete bipartite graph, a graph from $\mathcal{S}$ and a graph from $\mathcal{T}$, and it is* NP*-complete otherwise.*

Note that by the aforementioned result of Hickingbotham [48], we may replace "treewidth" by "path-width" in Theorem 20. However, currently, we know of only one problem that satisfies the conditions of Theorem 20, namely Edge Steiner Tree [13], which as we showed in our paper is also a C123-problem. Even though the conditions of Theorem 20 are very restrictive, we believe the following open problem is still interesting.

Open Problem 8 *Determine other graph problems that satisfy the conditions of Theorem 20.*

Acknowledgments. The authors wish to thank Hans Bodlaender, Daniel Lokshtanov and Vadim Lozin for helpful suggestions.

[6] The reason is that for any integer k and a sufficiently large integer ℓ, the class of subcubic $(C_3, \ldots, C_\ell, H_1, \ldots, H_k)$-free graphs coincides with the class of subcubic $(C_3, \ldots, C_\ell, H_1, \ldots, H_k)$-subgraph-free graphs.

References

1. Abboud, A., Williams, V.V., Wang, J.R.: Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. Proc. SODA 2016 pp. 377–391 (2016)
2. Abrishami, T., Alecu, B., Chudnovsky, M., Hajebi, S., Spirkl, S.: Induced subgraphs and tree-decompositions VII. Basic obstructions in H-free graphs. CoRR **2212.02737** (2022)
3. Abrishami, T., Chudnovsky, M., Vušković, K.: Induced subgraphs and tree decompositions I. Even-hole-free graphs of bounded degree. Journal of Combinatorial Theory, Series B **157**, 144–175 (2022)
4. Alekseev, V.E.: On easy and hard hereditary classes of graphs with respect to the independent set problem. Discrete Applied Mathematics **132**, 17–26 (2003)
5. Alekseev, V.E., Boliac, R., Korobitsyn, D.V., Lozin, V.V.: NP-hard graph problems and boundary classes of graphs. Theoretical Computer Science **389**, 219–236 (2007)
6. Alekseev, V.E., Korobitsyn, D.V.: Complexity of some problems on hereditary graph classes. Diskretnaya Matematika **4**, 34–40 (1992)
7. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. Journal of Algorithms **12**, 308–340 (1991)
8. Arnborg, S., Proskurowski, A.: Linear time algorithms for NP-hard problems restricted to partial k-trees. Discrete Applied Mathematics **23**, 11–24 (1989)
9. Bienstock, D., Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a forest. Journal of Combinatoral Theory, Series B **52**, 274–283 (1991)
10. Bodlaender, H.L.: On linear time minor tests with depth-first search. Journal of Algorithms **14**, 1–23 (1993)
11. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on Computing **25**, 1305–1317 (1996)
12. Bodlaender, H.L., Bonnet, É., Jaffke, L., Knop, D., Lima, P.T., Milanič, M., Ordyniak, S., Pandey, S., Suchý, O.: Treewidth is NP-hard on cubic graphs. Announcement during workshop at Lorentz Center (2022)
13. Bodlaender, H.L., Brettell, N., Johnson, M., Paesani, G., Paulusma, D., van Leeuwen, E.J.: Steiner trees for hereditary graph classes: A treewidth perspective. Theoretical Computer Science **867**, 30–39 (2021)
14. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) kernelization. Journal of the ACM **63**, 44:1–44:69 (2016)
15. Bodlaender, H.L., Gilbert, J.R., Hafsteinsson, H., Kloks, T.: Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. Journal of Algorithms **18**, 238–255 (1995)
16. Bodlaender, H.L., Hanaka, T., Kobayashi, Y., Kobayashi, Y., Okamoto, Y., Otachi, Y., van der Zanden, T.C.: Subgraph Isomorphism on graph classes that exclude a substructure. Algorithmica **82**, 3566–3587 (2020)
17. Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. Journal of Algorithms **21**, 358–402 (1996)
18. Boliac, R., Lozin, V.V.: On the clique-width of graphs in hereditary classes. Proc. ISAAC 2022, LNCS **2518**, 44–54 (2002)
19. Brook, R.L.: On colouring the nodes of a network. Mathematical Proceedings of the Cambridge Philosophical Society **37**, 194–197 (1941)
20. Bulatov, A.A.: A dichotomy theorem for nonuniform CSPs. Proc. FOCS 2017 pp. 319–330 (2017)
21. Chiarelli, N., Hartinger, T.R., Johnson, M., Milanic, M., Paulusma, D.: Minimum connected transversals in graphs: New hardness results and tractable cases using the price of connectivity. Theoretical Computer Science **705**, 75–83 (2018)
22. Chlebík, M., Chlebíková, J.: The complexity of combinatorial optimization problems on d-dimensional boxes. SIAM Journal on Discrete Mathematics **21**, 158–169 (2007)
23. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Mathematics **86**, 165–177 (1990)
24. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Information and Computation **85**, 12–75 (1990)
25. Dabrowski, K.K., Paulusma, D.: Clique-width of graph classes defined by two forbidden induced subgraphs. Computer Journal **59**, 650–666 (2016)
26. Dailey, D.P.: Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. Discrete Mathematics **30**, 289–293 (1980)
27. Darmann, A., Döcker, J., Dorn, B.: The Monotone Satisfiability problem with bounded variable appearances. International Journal of Foundations of Computer Science **29**, 979–993 (2018)
28. Dawar, A., Grohe, M., Kreutzer, S., Schweikardt, N.: Approximation schemes for first-order definable optimisation problems. Proc. LICS 2006 pp. 411–420 (2006)

29. Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. Journal of the ACM **52**, 866–893 (2005)
30. Deng, X., Lin, B., Zhang, C.: Multi-multiway cut problem on graphs of bounded branch width. Proc. FAW-AAIM 2013, LNCS **7924**, 315–324 (2013)
31. Diestel, R.: Graph Theory, 3rd edition. Springer (2005)
32. Ellis, J.A., Sudborough, I.H., Turner, J.S.: The vertex separation and search number of a graph. Information and Computation **113**, 50–79 (1994)
33. Evald, J., Dahlgaard, S.: Tight hardness results for distance and centrality problems in constant degree graphs. CoRR **abs/1609.08403** (2016)
34. Feghali, C., Lucke, F., Paulusma, D., Ries, B.: New hardness results for matching cut and disconnected perfect matching. CoRR **abs/2212.12317** (2022)
35. Fiorini, S., Hardy, N., Reed, B.A., Vetta, A.: Planar graph bipartization in linear time. Discrete Applied Mathematics **156**, 1175–1180 (2008)
36. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Excluded grid minors and efficient polynomial-time approximation schemes. Journal of the ACM **65**, 10:1–10:44 (2018)
37. Foucaud, F., Hocquard, H., Lajou, D.: Complexity and algorithms for injective edge-coloring in graphs. Inf. Process. Lett. **170**, 106121 (2021)
38. Garey, M.R., Johnson, D.S.: The Rectilinear Steiner Tree problem is NP complete. SIAM Journal of Applied Mathematics **32**, 826–834 (1977)
39. Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified NP-complete problems. Proc. STOC 1974 pp. 47–63 (1974)
40. Garey, M.R., Johnson, D.S., Tarjan, R.E.: The Planar Hamiltonian Circuit problem is NP-complete. SIAM Journal on Computing **5**, 704–714 (1976)
41. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., USA (1979)
42. Goldreich, O.: Computational complexity: A conceptual perspective. SIGACT News **39**, 35–39 (2008)
43. Golovach, P.A., Paulusma, D.: List coloring in the absence of two subgraphs. Discrete Applied Mathematics **166**, 123–130 (2014)
44. Golovach, P.A., Paulusma, D., Ries, B.: Coloring graphs characterized by a forbidden subgraph. Discrete Applied Mathematics **180**, 101–110 (2015)
45. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. Journal of the ACM **54**, 1:1–1:24 (2007)
46. Hadlock, F.: Finding a maximum cut of a planar graph in polynomial time. SIAM Journal on Computing **4**, 221–225 (1975)
47. Harary, F.: Graph theory. Addison-Wesley (1991)
48. Hickingbotham, R.: Induced subgraphs and path decompositions. CoRR **abs/2206.15054** (2022)
49. Jaffke, L., Kwon, O., Telle, J.A.: Mim-width i. induced path problems. Discrete Applied Mathematics **278**, 153–168 (2020)
50. Jansen, K., Scheffler, P.: Generalized coloring for tree-like graphs. Discrete Applied Mathematics **75**, 135–155 (1997)
51. Johnson, M., Martin, B., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Edge Multiway Cut and Node Multiway Cut are NP-hard on subcubic graphs. CoRR **abs/2211.12203** (2022)
52. Johnson, M., Martin, B., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Complexity framework for forbidden subgraphs: When problems are polynomial on subcubic graphs. Manuscript (2023)
53. Kamiński, M.: Max-Cut and containment relations in graphs. Theoretical Computer Science **438**, 89–95 (2012)
54. Kern, W., Paulusma, D.: Contracting to a longest path in H-free graphs. Proc. ISAAC 2020, LIPIcs **181**, 22:1–22:18 (2020)
55. Kinnersley, N.G.: The vertex separation number of a graph equals its path-width. Information Processing Letters **42**, 345–350 (1992)
56. Korhonen, T.: Grid induced minor theorem for graphs of small degree. CoRR **2203.13233** (2022)
57. Korpelainen, N., Lozin, V.V., Malyshev, D.S., Tiskin, A.: Boundary properties of graphs for algorithmic graph problems. Theoretical Computer. Science **412**, 3545–3554 (2011)
58. Kreutzer, S.: Algorithmic meta-theorems. London Mathematical Society Lecture Note Series **379**, 177–270 (2011)
59. Le, V.B., Telle, J.A.: The Perfect Matching Cut problem revisited. Proc. WG 2021, LNCS **12911**, 182–194 (2021)
60. Lozin, V.V., Razgon, I.: Tree-width dichotomy. European Journal of Combinatorics **103**, 103517 (2022)
61. Martin, B., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Complexity framework for forbidden subgraphs: When hardness is not preserved under edge subdivision. CoRR **abs/2211.14214** (2022)

62. Martin, B., Paulusma, D., Smith, S.: Hard problems that quickly become very easy. Information Processing Letters **174**, 106213 (2022)
63. Middendorf, M., Pfeiffer, F.: On the complexity of the Disjoint Paths problem. Combinatorica **13**, 97–107 (1993)
64. Mohar, B.: Face covers and the genus problem for apex graphs. Journal of Combinatorial Theory, Series B **82**, 102–117 (2001)
65. Monien, B., Sudborough, I.H.: Min Cut is NP-complete for edge weighted treees. Theoretical Computer Science **58**, 209–229 (1988)
66. Munaro, A.: Boundary classes for graph problems involving non-local properties. Theoretical Computer Science **692**, 46–71 (2017)
67. Nesetril, J., de Mendez, P.O.: Sparsity - Graphs, Structures, and Algorithms, Algorithms and combinatorics, vol. 28. Springer (2012)
68. Poljak, S.: A note on stable sets and colorings of graphs. Commentationes Mathematicae Universitatis Carolinae **15**, 307–309 (1974)
69. Robertson, N., Seymour, P.D.: Graph minors. III. Planar tree-width. Journal of Combinatorial Theory, Series B **36**, 49–64 (1984)
70. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. Journal of Combinatorial Theory, Series B **41**, 92–114 (1986)
71. Scheffler, P.: A practical linear time algorithm for disjoint paths in graphs with bounded tree-width. Preprint 396, Technische Universität Berlin, Institut für Mathematik (1994)
72. Seese, D.: Linear time computable problems and first-order descriptions. Mathematical Structures in Computer Science **6**, 505–526 (1996)
73. Speckenmeyer, E.: Untersuchungen zum Feedback Vertex Set Problem in ungerichteten Graphen. Ph.D. thesis, Paderborn (1983)
74. Telle, J.A., Proskurowski, A.: Algorithms for vertex partitioning problems on partial k-trees. SIAM Journal on Discrete Mathematics **10**, 529–550 (1997)
75. Ueno, S., Kajitani, Y., Gotoh, S.: On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. Discrete Mathematics **72**, 355–360 (1988)
76. Vatshelle, M.: New Width Parameters of Graphs. Ph.D. thesis, University of Bergen (2012)
77. Weißauer, D.: In absence of long chordless cycles, large tree-width becomes a local phenomenon. Journal of Combinatorial Theory, Series B **139**, 342–352 (2019)
78. Williams, R.: A new algorithm for optimal 2-constraint satisfaction and its implications. Theoretical Computer Science **348**, 357–365 (2005)
79. Williams, V.V.: On some fine-grained questions in algorithms and complexity. Proc. ICM 2018 pp. 3447–3487 (2018)
80. Xi, W., Lin, W.: On maximum P_3-packing in claw-free subcubic graphs. Journal of Combinatorial Optimization **41**, 694–709 (2021)
81. Yannakakis, M.: Node and edge deletion NP-complete problems. Proc. STOC 1978 pp. 253–264 (1978)
82. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. SIAM Journal on Applied Mathematics **38**, 364–372 (1980)
83. Zhuk, D.: A proof of the CSP dichotomy conjecture. Journal of the ACM **67**, 30:1–30:78 (2020)

A The Proof of Theorems 1 and 2

Both Theorems 1 and 2 follow immediately from the following classical result of Robertson and Seymour.

Theorem 21 ([70]). *For every planar graph H, all H-minor-free graphs have tree-width at most c_H for some constant c_H that only depends on the size of H.*

Here is the (known) proof of Theorem 1.

Theorem 1 (restated). *Let Π be a problem that is computationally hard on planar graphs, but efficiently solvable for every graph class of bounded treewidth. For any set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-minor-free graphs is efficiently solvable if $\mathcal{H}$ contains a planar graph and is computationally hard otherwise.*

Proof. For some integer $p \geq 1$, let $\mathcal{H}$ be a set of graphs, where we allow $\mathcal{H}$ to have infinite size. First assume that $\mathcal{H}$ contains a planar graph H_1. As G is $\mathcal{H}$-minor-free, G is H_1-minor-free. We now apply Theorem 21 to find that G has treewidth bounded by some integer c_{H_1}, which is a constant as H_1 is a fixed graph. We conclude that the class of $\mathcal{H}$-subgraph-free graphs has bounded treewidth. Hence, by our assumption on Π, we can solve Π efficiently for the class of $\mathcal{H}$-minor-free graphs.

Now assume that $\mathcal{H}$ contains no planar graph. As planar graphs are closed under taking vertex deletions, edge deletions and edge contractions, they are closed under taking minors. This means that the class of planar graphs is a subclass of the class of $\mathcal{H}$-minor-free graphs. Hence, by our assumption on Π, we find that Π is computationally hard for the class of $\mathcal{H}$-minor-free graphs. □

Here is the (known) proof of Theorem 2.

Theorem 2 (restated). *Let Π be a problem that is computationally hard on planar subcubic graphs, but efficienty solvable for every graph class of bounded treewidth. For any set of graphs $\mathcal{H}$, the problem Π on $\mathcal{H}$-topological-minor-free graphs is efficiently solvable if $\mathcal{H}$ contains a planar subcubic graph and is computationally hard otherwise.*

Proof. For some integer $p \geq 1$, let $\mathcal{H}$ be a set of graphs, where we allow $\mathcal{H}$ to have infinite size. First assume that $\mathcal{H}$ contains a planar subcubic graph H_1. As G is $\mathcal{H}$-topological-minor-free, G is H_1-topological-minor-free. As H_1 is subcubic, this means that G is even H_1-minor-free. We now apply Theorem 21 to find that G has treewidth bounded by some integer c_{H_1}, which is a constant as H_1 is a fixed graph. We conclude that the class of $\mathcal{H}$-subgraph-free graphs has bounded treewidth. Hence, by our assumption on Π, we can solve Π efficiently for the class of $\mathcal{H}$-topological-minor-free graphs.

Now assume that $\mathcal{H}$ contains no planar subcubic graph. As planar subcubic graphs are closed under taking vertex deletions, vertex dissolutions and edge deletions, they are closed under taking topological minors. This means that the class of planar subcubic graphs is a subclass of the class of $\mathcal{H}$-topological-minor-free graphs. Hence, by our assumption on Π, we find that Π is computationally hard for the class of $\mathcal{H}$-topological-minor-free graphs. □