Algorithmica
https://doi.org/10.1007/s00453-024-01289-2

Complexity Framework for Forbidden Subgraphs I: The Framework

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

Received: 28 November 2023 / Accepted: 23 December 2024

Abstract

For a set of graphs $\mathcal{H}$, a graph G is $\mathcal{H}$-subgraph-free if G does not contain any graph from $\mathcal{H}$ as a subgraph. We propose general and easy-to-state conditions on graph problems that explain a large set of results for $\mathcal{H}$-subgraph-free graphs. Namely, a graph problem must be efficiently solvable on graphs of bounded treewidth, computationally hard on subcubic graphs, and computational hardness must be preserved under edge subdivision of subcubic graphs. Our meta-classification says that if a graph problem Π satisfies all three conditions, then for every finite set $\mathcal{H}$, it is "efficiently solvable" on $\mathcal{H}$-subgraph-free graphs if $\mathcal{H}$ contains a disjoint union of one or more paths and subdivided claws, and Π is "computationally hard" otherwise. We apply our *meta-classification* on many well-known partitioning, covering and packing problems, network design problems and width parameter problems to obtain a dichotomy between polynomial-time solvability and NP-completeness. For distance-metric problems, we obtain a dichotomy between almost-linear-time solvability and having no subquadratic-time

✉ Daniël Paulusma
daniel.paulusma@durham.ac.uk

Matthew Johnson
matthew.johnson2@durham.ac.uk

Barnaby Martin
barnaby.d.martin@durham.ac.uk

Jelle J. Oostveen
j.j.oostveen@uu.nl

Sukanya Pandey
s.pandey1@uu.nl

Siani Smith
siani.smith@durham.ac.uk

Erik Jan van Leeuwen
e.j.vanleeuwen@uu.nl

1 Department of Computer Science, Durham University, Durham, UK

2 Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

algorithm (conditioned on some hardness hypotheses). Apart from capturing a large number of explicitly and implicitly known results in the literature, we also prove a number of new results. Moreover, we perform an extensive comparison between the subgraph framework and the existing frameworks for the minor and topological minor relations, and pose several new open problems and research directions.

Keywords Forbidden subgraph · Complexity dichotomy · Meta-classification · Treewidth

1 Introduction

Algorithmic meta-theorems are general algorithmic results applying to a whole range of problems, rather than just a single problem alone [73]. 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 [31], 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 Sect. 3 for a definition of treewidth). Another example is that of Seese [90], 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. [36], 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, [16, 38, 45]), 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 problem is "efficiently solvable"; otherwise it is "computationally hard". Throughout, we let these two notions depend on context; for example, efficiently solvable and computationally hard could mean being solvable in polynomial time and being NP-complete, respectively.

Algorithmic meta-classifications are less common than algorithmic meta-theorems, but let us mention two famous results. Grohe [57] 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 modulo homomorphic equivalence (assuming $\mathrm{W}[1] \neq \mathrm{FPT}$). Bulatov [25] and Zhuk [105] proved that every finite-domain $\mathrm{CSP}(H)$ is either polynomial-time solvable or NP-complete, omitting any Ladner-like complexities in between.

Two well-known meta-classifications apply to the classes of $\mathcal{H}$-minor-free graphs and $\mathcal{H}$-topological-minor-free graphs. For a set $\mathcal{H}$ of graphs, these are the class of graphs G where, starting from G, no graph $H \in \mathcal{H}$ can be obtained by a series of vertex deletions, edge deletions, and edge contractions, respectively a series of vertex deletions, edge deletions, and vertex dissolutions (see Sect. 2 for full definitions). Both are a consequence of a classic result of [87]; we refer to Appendix A for proof details, but see also e.g. [72].

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 (or equivalently, if the class of* $\mathcal{H}$*-minor-free graphs has bounded treewidth) 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 (or equivalently, if the class of* $\mathcal{H}$*-topological-minor-free graphs has bounded treewidth) and is computationally hard otherwise.*

Later in our paper, we will discuss many problems that satisfy the conditions of Theorems 1 and 2. We refer, for example, to [46, 82] 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.

On the other end of the spectrum lie the classes of $\mathcal{H}$-free graphs (or hereditary graph classes). A graph G is $\mathcal{H}$-free if, starting from G, no graph $H \in \mathcal{H}$ can be obtained by a series of vertex deletions. Hereditary graph classes are much more complex in structure than the classes of $\mathcal{H}$-minor-free graphs and $\mathcal{H}$-topological-minor-free graphs, for which powerful structure theorems exist [58, 88]. However, there exist many infinite antichains under the induced subgraph relation (e.g. the set of cycles) that are not antichains under the minor and topological minor relations. This makes the task of finding algorithmic meta-classifications much harder. In fact, even algorithmic meta-theorems are difficult to obtain for the induced subgraph relation, even for a single forbidden graph H. Indeed, complexity dichotomies for H-free graphs are rare and only known for specific problems (see e.g. [15, 55, 67, 68]).

Despite the above, some attempts have been made to study complexity boundaries, e.g. through the notion of boundary graph classes [5] (see also [6, 72, 82]). However, the induced subgraph relation is far from being understood. For example, after more than forty years of research on Independent Set for H-free graphs starting from the work of Alekseev [5], currently only a trichotomy is known between being polynomial-time solvable, quasi-polynomial-time solvable and NP-complete (see the recent work of Gartland et al. [52]). We do not yet know how to obtain the dichotomy between polynomial-time solvable and NP-complete we believe this implies (see [59] for the most recent progress). Many other fundamental problems are still far from being settled for H-free graphs with infinitely many open cases even when H is a connected graph.

Between $\mathcal{H}$-minor-free graphs and $\mathcal{H}$-topological-minor-free graphs on the one side and $\mathcal{H}$-free graphs on the other side, lies the class of *$\mathcal{H}$-subgraph-free graphs*. These are the graphs G where, starting from G, no graph $H \in \mathcal{H}$ can be obtained by a series of vertex or edge deletions. In general, for every set $\mathcal{H}$ of graphs, the following holds (see also Fig. 1 for some small examples):

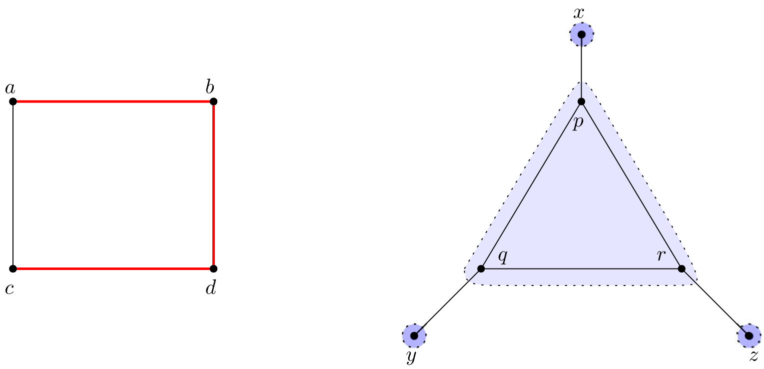

Fig. 1 The left example shows that the C_4 is not P_4-subgraph-free (the red edges correspond to a P_4 subgraph), but it is P_4-free. The right example shows that the net is not $K_{1,3}$-minor-free (the vertex sets indicated in blue correspond to a $K_{1,3}$ minor), but it is $K_{1,3}$-topological-minor-free (Color figure online)

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

Forbidden subgraphs represent many rich graph classes. To explain this, let C_r, K_r and P_r denote the path, complete graph and cycle on n vertices, respectively, and let $K_{p,q}$ denote the complete bipartite graph whose two partition classes each have size p and q, respectively. It is readily seen that, for example:

- The classes of graphs of maximum degree at most r and $K_{1,r+1}$-subgraph-free graphs coincide;
- The class of graphs with girth larger than g for some integer $g \geq 3$ coincides with the class of $(C_3, \ldots, C_g)$-subgraph-free graphs (and with the class of $(C_3, \ldots, C_g)$-free graphs);
- A class of graphs $\mathcal{G}$ has bounded treedepth if it is a subclass of P_r-subgraph-free graphs for some constant r, and vice versa [84]; and
- For every class $\mathcal{G}$ of degenerate or nowhere dense graphs [83], there exists an integer t such that every $G \in \mathcal{G}$ is $K_{t,t}$-subgraph-free (see [95] for a proof).

Moreover, H-free graphs and H-subgraph-free graphs coincide if and only if $H = K_r$ for some integer $r \geq 1$. This leads to a rich structural landscape.

A substantial body of work has studied the parameterized complexity of graph problems on a restricted set of subgraph-free graph classes (notably through the lens of sparsity, see e.g. [91]). However, $\mathcal{H}$-subgraph-free graphs have been significantly less studied in the context of classical complexity theory than the other classes, despite capturing many natural graph classes. This warrants a more in-depth look at $\mathcal{H}$-subgraph-free graphs.

Adding to this, $\mathcal{H}$-subgraph-free graphs seem to exhibit extreme and unexpected jumps in problem complexity. For example, there exist problems that are PSPACE-complete in general but constant-time solvable for every $\mathcal{H}$-free graph class [78] 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, which is to decide for a given integer k and graph G, if G contains a *clique* (set of pairwise adjacent vertices) of size at least k. The Clique problem is well-known to be NP-hard. (see [51]). However, for $\mathcal{H}$-subgraph-free graphs, the situation drastically changes. The reason is that the size of a largest clique is bounded by the number of vertices of a smallest graph in $\mathcal{H}$ and hence, one can just apply brute force to find a largest clique in an $\mathcal{H}$-subgraph-free graph in polynomial time. Hence, the following holds.

Observation 1 *For every set of graphs $\mathcal{H}$,* Clique *is polynomial-time solvable for $\mathcal{H}$-subgraph-free graphs.*

In contrast to $\mathcal{H}$-free graphs, some work has pointed to more complex dichotomy results being possible. Kamiński [67] gave a complexity dichotomy 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 [7] did the same for Independent Set, Dominating Set and Long Path; see [56] for a short, alternative proof (similar to the one of [67] for Max- Cut) for the classification for Independent Set for H-subgraph-free graphs. In [55] the computational complexity of List Colouring for $\mathcal{H}$-subgraph-free graphs has been determined for every finite set of graphs $\mathcal{H}$. More recently, Bodlaender et al. [17] determined the computational complexity of Subgraph Isomorphism for H-subgraph-free graphs for all connected graphs H except the case where $H = P_5$, and they reduced all open "disconnected" cases to either $H = P_5$ or $H = 2P_5$. However, even for a classical problem such as Colouring, a complete complexity classification for H-subgraph-free graphs is far from settled [56]. Many more problems have not been studied in this context at all.

Our Focus. Motivated by our apparent lack of understanding of $\mathcal{H}$-subgraph-free graphs, we embark on a deeper investigation of the computational complexity of graph problems restricted to $\mathcal{H}$-subgraph-free graphs. In this way, we will pioneer a new meta-classification of $\mathcal{H}$-subgraph-free graphs, which is only the third meta-classification for graph containment apart from Theorems 1 and 2. Besides the aforementioned complexity dichotomies from [7, 56, 67], we will show that many other problems are covered by this meta-classification. To do this, we will survey and apply known results from the literature and also prove new results.

1.1 The Meta-Classification for $\mathcal{H}$-subgraph-free graphs

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. For an integer $k \geq 1$, the *k-subdivision* of an edge $e = uv$ of a graph replaces e by a path of length $k + 1$ with endpoints u and v (and k new vertices). The *k-subdivision* of a graph G is the graph obtained from G after k-subdividing each edge (see Fig. 2 for an example of a 2-subdivision). For a graph class $\mathcal{G}$ and an integer k, let $\mathcal{G}^k$ consist of the k-subdivisions of the graphs in $\mathcal{G}$.

Note that a problem that is hard for a class $\mathcal{G}$ may be efficiently solvable for infinitely many $\mathcal{G}^p$, while it may be computationally hard for infinitely many other $\mathcal{G}^p$ (see also Sect. 6.2). In order to capture this behaviour appropriately we introduce the following condition. A graph problem Π is computationally hard *under edge subdivision*

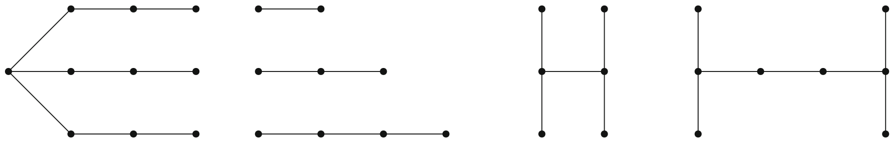

Fig. 2 Left: An example of a graph in $\mathcal{S}$ (the graph $S_{3,3,3} + P_2 + P_3 + P_4$); also note that $S_{3,3,3}$ is the 2-subdivision of $K_{1,3}$. Right: the graphs $\mathbb{H}_1$ and $\mathbb{H}_3$, where $\mathbb{H}_1$ is the "H"-graph, formed by an edge (the *middle edge*) joining the middle vertices of two P_3s, and $\mathbb{H}_i$ ($i \geq 2$) is obtained from $\mathbb{H}_1$ by $(i-1)$-subdividing the middle edge

of subcubic graphs if for every integer $j \geq 1$, there is an integer $\ell \geq j$ such that: if Π is computationally hard for the class $\mathcal{G}$ of subcubic graphs, then Π is computationally hard for $\mathcal{G}^{\ell}$. Commonly, we can prove that the condition holds by showing that computational hardness is maintained under k-subdivision for a small integer k (e.g. $k = 1, 2, 3, 4$) and then repeatedly apply the k-subdivision operation.

Our framework contains every graph problem Π satisfying the following three conditions (recall that the notions of efficiently solvable and computational hardness depend on their context):

C1. Π is efficiently solvable for every graph class of bounded treewidth;
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*. Note that if a problem does not satisfy C2, then C3 is implied. As mentioned, we refer to Sect. 6.2 for some reasons why we cannot simplify condition C3. In the same section we also explain why being subcubic is important in condition C3.

For some $p, q, r \geq 1$, the *subdivided* claw $S_{p,q,r}$ is obtained from the *claw* (the 4-vertex star $K_{1,3}$) after $(p-1)$-, $(q-1)$-, and $(r-1)$-subdividing its three edges respectively. The *disjoint union* $G_1 + G_2$ of two vertex-disjoint graphs G_1 and G_2 is the graph $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$. We now define the set $\mathcal{S}$, which is well-known in the literature (see, for example, [6, 15, 76]) and also plays an important role in our paper; see the left side of Fig. 2 for an example of a graph that belongs to $\mathcal{S}$.

Definition 1 The set $\mathcal{S}$ consists of all non-empty disjoint unions of zero or more subdivided claws and paths.

Our main result is the following theorem that can be seen as the "subgraph variant" of Theorems 1 and 2. Note that it suggests, just like Theorems 1 and 2, that boundedness of treewidth might be the underlying explanation for the polynomial-time solvability.

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 $\mathcal{S}$ (or equivalently, if the class of $\mathcal{H}$-subgraph-free graphs has bounded treewidth) and computationally hard otherwise.*

We prove Theorem 3 in Sect. 3. The proof of the "efficient" part of Theorem 3 uses a well-known path-width result [11]. To prove the "hard" part of Theorem 3, we show that every problem satisfying C2 and C3 is hard for $(C_3, \ldots, C_i, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_i)$-subgraph-free graphs for every integer $i \geq 3$ (see the right side of Fig. 2 for some examples of a subdivided "H"-graph $\mathbb{H}_i$). As such, the proof is similar to the proofs for Max- Cut [67] and List Colouring [55] and Independent Set for H-subgraph-free graphs [56] for finite sets of graphs $\mathcal{H}$, which as we will see are all C123-problems. The original proofs from Alekseev and Korobitsyn [7] for Independent Set, Dominating Set and Long Path, restricted to $\mathcal{H}$-free graphs for finite sets of graphs $\mathcal{H}$, are different and do not involve any direct path-width arguments.

1.2 Impact

The impact of the subgraph framework is three-fold. These impacts follow from the broad overview of the literature provided in this paper on problems that exhibit zero or more of the properties C1, C2, C3.

First Impact. First and foremost, we are able to provide a complete dichotomy for many problems on $\mathcal{H}$-subgraph-free graphs by showing they are C123-problems. In this way, we obtain a dichotomy between polynomial-time solvability and NP-completeness for many well-known partitioning, covering and packing problems, network design problems and width parameter problems.

We first show in Sect. 4 that computing the path-width and treewidth of a graph are C123-problems. We do the same for a number of 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 [61]. We also show that Independent Set (or equivalently, Vertex Cover) and Dominating Set are C123, and thus we recover the known classifications of [7]. Moreover, we show that List Colouring is C123, and thus we re-obtain the classification of [55].

Next we prove (still in Sect. 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 recover the classification of [7] for Long Path restricted to $\mathcal{H}$-subgraph-free graphs. We also include a reference to a subsequent result of [43], in which it was shown that Perfect Matching Cut is C123.

We then consider in Sect. 5, the polynomial-time solvable problems Diameter and Radius. These problems 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 [99] and Hitting Set Conjecture [1], respectively. The Orthogonal Vectors conjecture is implied by the Strong Exponential Time Hypothesis (SETH) [99] and by the Hitting Set Conjecture [1]; we refer to [100] for more context on both conjectures.

The above applications of Theorem 3, as well as a number of applications of Theorem 1 and 2, are summarized in Table 1. A detailed comparison is deferred to Sect. 7.

Table 1 The minor framework (MF), topological minor framework (TMF), and subgraph framework (SF), with the conditions.

Graph problem	MF	TMF	SF	Plan.	Subc. plan.	C1	C2	C3
Path-Width	✓	✓	✓	[81]	[81]	[19]	[81]	T7
Tree-Width	?	?	✓	?	?	[13]	[14]	T8
Dominating Set	✓	✓	✓	[51]	[51]	[9]	[51]	[27]
Independent Dominating Set	✓	✓	✓	[30]	[30]	[94]	[30]	[27]
Edge Dominating Set	✓	✓	✓	[104]	[104]	[8]	[104]	[27]
Independent Set	✓	✓	✓	[80]	[80]	[9]	[80]	[85]
Vertex Cover	✓	✓	✓	[80]	[80]	[9]	[80]	[85]
Connected Vertex Cover	✓	✗	✗	[48]	no	[8]	no	triv
Feedback Vertex Set	✓	✗	✗	[92]	no	[8]	no	triv
Independent Feedback Vertex Set	✓	✗	✗	[92]	no	[93]	no	triv
Odd Cycle Transversal	✓	✓	✓	T11	T11	[8]	T11	T11
Independent Odd Cycle Transv.	✓	✓	✓	T11	T11	[8]	T11	T11
C_5-Colouring	✓	✓	✗	[77]	T22	[37]	[47]	no
3-Colouring	✓	✗	✗	[54]	no	[9]	no	triv
Star 3-Colouring	✓	✓	✗	[4]	[20]	[31]	[20]	no
List Colouring	✓	✓	✓	[64]	T11	T11	T11	T11
P_3-Factor	✓	✓	✓	[102]	[102]	[6]	[6]	[6]
Edge Steiner Tree	✓	✓	✓	[48]	T15	[8]	T15	T15

Table 1 continued

Graph problem	MF	TMF	SF	Plan.	Subc. plan.	C1	C2	C3
Node Steiner Tree	✓	✓	✓	[48]	T15	[8]	T15	T15
Steiner Forest	✗	✗	✗	[48]	T15	no	T15	T15
Disjoint Paths	✓	✓	✓	[79]	[79]	[89]	[79]	T17
Induced Disjoint Paths	✓	✓	✓	T17	T17	[63]	T17	T17
Long Cycle	✓	✓	✓	[49]	[49]	[12]	[49]	T18
Long Induced Cycle	✓	✓	✓	[49]	[49]	[63]	T18	T18
Hamilton Cycle	✓	✓	✗	[49]	[49]	[9]	[50]	no
Long Path	✓	✓	✓	[49]	[49]	[12]	[49]	T18
Long Induced Path	✓	✓	✓	T18	T18	[63]	T18	T18
Hamilton Path	✓	✓	✗	[49]	[49]	[9]	[50]	no
Max- Cut	✗	✗	✓	no	no	[8]	[103]	[67]
Edge Multiway Cut	✓	✓	✓	[65]	[65]	[39]	[65]	T16
Node Multiway Cut	✓	✓	✓	[65]	[65]	[8]	[65]	T16
Matching Cut	✓	✗	✗	[23]	no	[23]	no	triv
Perfect Matching Cut	✓	✓	✓	[22]	[22]	[74]	[74]	[43]
Diameter	✗*	✗*	✓	no	no	[1]	[42]	L1
Radius	✗*	✗*	✓	no	no	[1]	[42]	T19
Subgraph Isomorphism	✗	✗	✗	[17]	[17]	no	[17]	triv
Clique	✗	✗	✗	no	no	triv	no	triv

The problems are mainly chosen to illustrate the wide reach of the frameworks and their differences. If a problem satisfies the conditions of a meta-classification, we indicate this with ✓; if not, with a ✗; and if unknown with a "?". A reference in a column is a reference to where the condition is shown to hold, "triv" means that the condition holds trivially, and "no" means the condition does not hold. A further discussion of the table, explaining the "no"-statements, is in Sect. 7. *For Diameter and Radius, the complexity on (subcubic) planar graphs is still open; however, a distinction almost-linear versus not subquadratic (as in Theorem 19) is not possible [53]. Recall that if a problem does not satisfy C2, then C3 is implied

Second Impact. The second impact of our framework is that we uncover several complexity gaps in the literature. We subsequently resolve them in this and other work. First, an important and difficult open question turned out to be the complexity of Edge/Node Multiway Cut, for which the classic results of Dahlhaus et al. [33] shows NP-completeness of the unweighted variant for planar graphs of maximum degree 11 (and claims an improved bound of 6). In a companion paper [65], we show NP-completeness for Edge/Node Multiway Cut on planar subcubic graphs, besting the earlier degree bound and showing these problems satisfy C2. We note that the NP-completeness for planar subcubic graphs instead of only for subcubic graphs is also helpful in proving that a problem belongs to the minor and topological minor frameworks.

Second, we prove, as new results, that List Colouring, Odd Cycle Transversal, Independent Odd Cycle Transversal, and C_5- Colouring are NP-complete for planar subcubic graphs, and thus satisfy C2. For the following problems we give explicit proofs to show that they satisfy C3: Path- Width, Tree- Width, Edge/Node Steiner Tree, Edge/Node Multiway Cut and Diameter. Hence, our framework on $\mathcal{H}$-subgraph-free graphs shows the way towards new results.

Third Impact. The third impact of our framework is that it enables a structured investigation into complexity dichotomies for graph problems that do not satisfy some of the conditions, C1, C2 or C3, particularly when only one is not satisfied. We call such problems C23, C13, or C12, respectively. This direction leads to many interesting new research questions. We are currently trying to determine new complexity classifications for a number of relevant problems in follow-up work that includes three papers labelled as *Complexity Framework For Forbidden Subgraphs* II [75], III [66] and IV [18]. We consider C12-problems in [75]; C13-problems in [66]; and C23-problems in [18]. This led to new insights into the complexity of well-studied problems such as the C12-problem Hamilton Cycle [75], the C13-problem Feedback Vertex Set [66], and the C23-problem Steiner Forest [18]. For all these problems and several more [18, 66, 75], the complexity classifications are currently incomplete and will be different from the one in Theorem 3. Hence, our framework has the potential to open a new and rich research area.

1.3 Organization

We start with some preliminaries in Sect. 2. We prove Theorem 3 as a consequence of a stronger result In Sect. 3. Next, in Sects. 4 and 5, we apply our subgraph framework to a wealth of problems as described above. We provide a discussion on limitations of the framework in Sect. 6, and an extensive comparison of the applicability of the three meta-classifications (Theorem 1, 2, and 3) in Sect. 7. Finally, we conclude our paper with a list of open problems and research directions in Sect. 8.

2 Preliminaries

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 *girth* of a graph G that is not a forest is the length (number of edges) of a shortest cycle in G.

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 and call the operation the *vertex dissolution* of 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 $\mathcal{H} = \{H_1, \ldots, H_p\}$ for some integer $p \geq 1$, we also say that G is $(H_1, \ldots, H_p)$*-subgraph-free*. 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 said to be *hereditary*.

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 [40]). Hence, treewidth is more powerful than path-width.

3 The Proof of Theorem 3

We present a stronger result that will imply Theorem 3. A graph class closed under edge deletion is also called *monotone* [6, 21, 72]. For a set of graphs $\mathcal{H}$, the class of $\mathcal{H}$-subgraph-free graphs is *finitely defined* if $\mathcal{H}$ is a finite set. We say that a problem Π is C1′D if Π satisfies the following two conditions (see Fig. 2 for examples of the subdivided "H"-graphs $\mathbb{H}_i$):

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

D. For every $i \geq 3$, Π is computationally hard for the class of $(C_3, \ldots, C_i, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_i)$-subgraph-free graphs.

Our first theorem shows that the class of C1′D-problems is a proper superclass of the class of C123-problems.

Theorem 4 *Every C123-problem is C1′D, but not every C1′D-problem is C123.*

Proof Let Π be a C123-problem. Then Π satisfies C1 and thus C1′. To show condition D, let $i \geq 3$, and let $\mathcal{G}_i$ be the class of $(C_3, \ldots, C_i, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_i)$-subgraph-free graphs. As Π satisfies C2, Π is computationally hard for the class $\mathcal{G}$ of subcubic graphs, that is, $K_{1,4}$-subgraph-free graphs. As Π satisfies C3, there exists an integer $\ell \geq i + 1$, such that Π is computationally hard for $\mathcal{G}^\ell$. We note that $\mathcal{G}^\ell$ is a subclass of $\mathcal{G}_i$. Hence, Π is computationally hard for $\mathcal{G}_i$ and thus satisfies D. We conclude that Π is a C1′D-problem.

To show that the reverse statement does not necessarily hold, we define the following (artificial) example problem. Let $\mathcal{B}$ be the set of all graphs obtained from a cycle after adding a new vertex made adjacent to precisely one vertex of the cycle. Then the problem $\mathcal{B}$- Modified List Colouring takes as input a graph G with a list assignment L and asks whether G simultaneously has a colouring respecting L and has a connected component that is a graph from $\mathcal{B}$.

We now prove that $\mathcal{B}$- Modified List Colouring is not C123 but is C1′D. We distinguish between "being polynomial-time solvable" and "being NP-complete". We first observe that $\mathcal{B}$ satisfies 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 $\mathcal{B}$.

Due to Property 1, $\mathcal{B}$- Modified List Colouring does not satisfy C3. Hence, $\mathcal{B}$- Modified List Colouring is not a C123-problem. We will prove that $\mathcal{B}$- Modified List Colouring is C1′D. As List Colouring is C123 by Theorem 11, it satisfies C1 and thus C1′. 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 satisfies C1′. Below we prove that it also satisfies condition D.

Let $i \geq 3$, and let $\mathcal{G}_i$ be the class of $(C_3, \ldots, C_i, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_i)$-subgraph-free graphs. As List Colouring is C123, it follows from the first statement that it is also C1′D. Hence, List Colouring is NP-complete on $\mathcal{G}_i$. Let (G, L) be an instance of List Colouring where G is a graph from $\mathcal{G}_i$. We note that $\{C_3, \ldots, C_i, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_i)\} \cap \mathcal{S} = \emptyset$. Hence, 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. We conclude that $\mathcal{B}$- Modified List Colouring satisfies D and is thus a C1′D-problem. As we already showed that $\mathcal{B}$- Modified List Colouring is not C123, this proves the second statement of the theorem. □

We also need a theorem from Bienstock, Robertson, Seymour and Thomas.

Theorem 5 *([11]) For every forest F, all F-minor-free graphs have path-width at most $|V(F)| - 2$.*

We now prove a result, which shows that the conditions C1′ and D are both necessary and sufficient.

Theorem 6 *Let Π be a problem. Then the following two statements are equivalent:*

(i) Π is C1′D; and
(ii) 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 First assume that Π is C1′D. Let $\mathcal{H}$ be a finite set of graphs. First suppose that $\mathcal{H}$ contains a graph H from $\mathcal{S}$. Let G be a $\mathcal{H}$-subgraph-free graph. As G is $\mathcal{H}$-subgraph-free, G is H-subgraph-free. It is known (see e.g. [55, 56]) that, for any graph $H' \in \mathcal{S}$, a H'-subgraph-free graph is also H'-minor-free. Hence, G is H-minor-free. So by Theorem 5, G has constant path-width at most $|V(H)| - 2$, meaning we can solve Π efficiently by C1′.

Now suppose that $\mathcal{H}$ contains no graph from $\mathcal{S}$. Let $H \in \mathcal{H}$. As $H \notin \mathcal{S}$, H has a connected component containing a $K_{1,4}$ (or equivalently, a vertex of degree at least 4); or a cycle C_h for some $h \geq 3$; or a graph $\mathbb{H}_i$ for some $i \geq 1$. Hence, the class of H-subgraph-free graphs contains the $K_{1,4}$-subgraph-free graphs; or C_h-subgraph-free graphs for some $h \geq 3$; or $\mathbb{H}_i$-subgraph-free graphs for some $i \geq 1$, each of which contains the $(C_3, \ldots, C_{j(H)}, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_{j(H)})$-subgraph-free graphs, where $j(H) = \max\{h, i\}$. Hence, the class of H-subgraph-free graphs contains the $(C_3, \ldots, C_{j(H)}, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_{j(H)})$-subgraph-free graphs. Consequently, the class of $\mathcal{H}$-subgraph-free graphs contains the $(C_3, \ldots, C_{j^*}, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_{j^*})$-subgraph-free graphs, where $j^* = \max_{H \in \mathcal{H}} j(H)$ (note that j exists, as $\mathcal{H}$ is finite). As Π satisfies D, we find that Π is computationally hard for $(C_3, \ldots, C_{j^*}, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_{j^*})$-subgraph-free graphs, and thus for $\mathcal{H}$-subgraph-free graphs.

Now assume that 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. We first prove C1′. Let $\mathcal{H}$ be a finite set, such that the class of $\mathcal{H}$-subgraph-free graphs has bounded path-width. Recall that the latter holds if and only if $\mathcal{H}$ contains a graph from $\mathcal{S}$ [86]. Hence, Π satisfies C1′.

We now prove that condition D holds. Let $i \geq 3$, and let $\mathcal{G}_i$ be the class of $\mathcal{H}$-subgraph-free graphs, where $\mathcal{H} = \{C_3, \ldots, C_i, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_i\}$. Then $\mathcal{H}$ contains no graph from $\mathcal{S}$. Hence, Π is computationally hard for $\mathcal{H}$-subgraph-free graphs. Consequently, Π satisfies D. □

We are now ready to prove Theorem 3, which we restate below.

Theorem 3 (restated). *Let Πbe 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 $\mathcal{S}$ (or equivalently, if the class of $\mathcal{H}$-subgraph-free graphs has bounded treewidth) and computationally hard otherwise.*

Proof The result follows from combining Theorems 4 and 6, and the well-known fact that for a finite set of graphs $\mathcal{H}$, a class of $\mathcal{H}$-subgraph-free graphs has bounded path-width if and only if it has bounded treewidth if and only if $\mathcal{H}$ contains a graph from $\mathcal{S}$ [86] (see e.g. [21, 32], for an explanation with respect to the more powerful parameter clique-width, and hence, replacing "bounded pathwidth" in C1′ by "bounded treewidth" or "bounded clique-width" yields the same equivalence as in Theorem 6). □

Remark. We emphasize that we are not aware of any natural C1′D-problem that is not C123. As the conditions C1–C3 are more intuitive, we have therefore chosen to present our subgraph framework in terms of the C1–C3 conditions instead of the C1′-D conditions.

4 Application to NP-Complete Problems

We provide a complete dichotomy between polynomial-time solvability and NP-completeness for many problems on $\mathcal{H}$-subgraph-free graphs by showing they are C123-problems. In Sect. 4.1, we give examples of width parameter problems that are C123. In Sect. 4.2 we give examples of partitioning, covering and packing problems that are C123. In Sect. 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 the relevant NP-completeness result applies to subcubic planar graphs, as reflected in Table 1. 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. We observe that it is unclear whether a path-width bound is maintained under subdivision, and thus proving property C3 for Path- Width is non-trivial. We show a more specific result that is sufficient for our purposes. For Tree- width, we can follow a more direct proof.

Theorem 7 PATH- WIDTH *is a C123-problem.*

Proof PATH- WIDTH is linear-time solvable for every graph class of bounded treewidth [19] so satisfies C1. It is NP-complete for 2-subdivisions of planar cubic graphs [81] 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.

First suppose that $G = (V, E)$ that is a 2-subdivision of a graph G'' has path-width k. We use the known equivalence of path-width to the vertex separation number [69]. 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 $\mathrm{vs}_L(G) = \max_{i\in\{1,\ldots,|V|\}}\{|V_L(i)|\}$ and $\mathrm{vs}(G) = \min_L\{\mathrm{vs}_L(G)\}$ is the *vertex separation number* of G.

As shown by Kinnersley [69], G has a layout L such that $\mathrm{vs}_L(G) = k$ since G has path-width k. In a 2-subdivision, such as G, any edge uv of the original graph (G'' in this case) 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 $L(u) < L(a)$ for each such edge uv of G''. By applying the transformation of Ellis, Sudborough and Turner [41, Lemma 2.3] if necessary, we may assume that L is a standard layout and still $\mathrm{vs}_L(G) = k$.

For some edge uv of G'' and its 2-subdivision into ua, ab, bv in G, consider a further subdivision of each of these three edges. 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 a in L and now a follows after x in L'. Hence, $V_{L'}(L'(x))$ has the same size as $V_L(L(a))$, so size 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 $V_L(L(b))$ with b replaced by z if $b \in V_L(L(b))$. Similarly, for any $i > L(b)$, if $b \in V_L(i)$, we can replace b by z to obtain $V_{L'}(i)$; otherwise, $V_{L'}(i) = V_L(i)$. Note that a is never part of $V_L(i)$ for $i > L(b)$. 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 [69], G' has path-width at most k.

Now suppose that G' has path-width k. As subdivision cannot decrease path-width (or consider the converse, contraction cannot increase it), G has path-width at most k.

From the above, we conclude that G has path-width k if and only if G' has path-width k. Hence, the claim, and thus C3, 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 [13]. Very recently, it was shown that TREE- WIDTH is NP-complete for cubic graphs [14] so the problem satisfies C2. It also satisfies C3, which is a well-known observation; see [101] for an explicit proof (see also, for example, [70]). Hence, the theorem follows. □

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 [7].

Theorem 9 VERTEX COVER *and* INDEPENDENT SET *are C123-problems.*

Proof Both are linear-time solvable for graphs of bounded treewidth [9] so satisfy C1. Both are NP-complete for 2-connected cubic planar graphs [80] 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 $k + m$ [85]. □

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 [61].

The following theorem shows that both problems are C123, just like DOMINATING SET; hence, we recover the classification of [7] for the latter problem.

Theorem 10 DOMINATING SET*,* INDEPENDENT DOMINATING SET *and* EDGE DOMINATING SET *are C123-problems.*

Proof DOMINATING SET [9], INDEPENDENT DOMINATING SET [94] and EDGE DOMINATING SET [8] are linear-time solvable for graphs of bounded treewidth so satisfy C1. DOMINATING SET [51], INDEPENDENT DOMINATING SET [30] and EDGE DOMINATING SET [104] are NP-complete for planar subcubic graphs so satisfy C2. For showing C3 we use the following claim (see for example [27]) 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). Indeed, because of Brooks' theorem [24], it is polynomial-time solvable on subcubic graphs. 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 [55] 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 [64] 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 of bounded treewidth [8, 44] 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 from Proposition 2.27 of [54]). 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 [35]). 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.

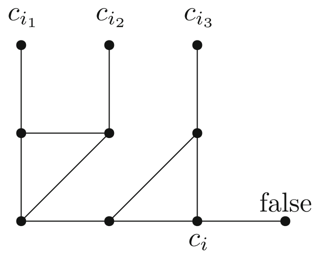

Fig. 3 The clause gadget that was also used in the reduction from (PLANAR) 3-SATISFIABILITY to 3-COLOURING, drawn with an edge connecting the output node to a vertex of the path P representing false. The property that the gadget enforces is that not all of the three input nodes $c_{i_1}, c_{i_2}, c_{i_3}$ may be coloured the same as the vertex representing false

- 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 true) 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.

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 alternatingly with 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 $S \subseteq V(T_1) \cup \cdots \cup V(T_{2m})$. Since we created a proper colouring, S consists of exactly one vertex of each T_i and its vertices are pairwise non-adjacent. 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 show C2 for Independent Odd Cycle Transversal and C3 for Odd Cycle Transversal and Independent Odd Cycle Transversal. To show C3 for Odd Cycle Transversal, we can just use the following claim (see for example [26] 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 prove 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 [55], 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 [6].

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

Proof This follows from combining Proposition 1 of [6] for showing C1 with Lemma 12 of [6] for showing C2 and C3 (with $k = 3$). Recently, Xi and Lin [102] 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 [67].

Theorem 13 *([67])* Max- Cut *is a C123-problem.*

Proof Max- Cut is linear-time solvable for graphs of bounded treewidth [8] and NP-complete for subcubic graphs [103] so satisfies C1 and C2. A cut C of a graph

G is *maximum* if G has no cut of greater size. Kamiński [67] 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. [43] recently showed that PERFECT MATCHING CUT is C123.

Theorem 14 *([43])* PERFECT MATCHING CUT *is a C123-problem.*

Proof Le and Telle [74] observed that PERFECT MATCHING CUT is polynomial-time solvable for graphs of bounded treewidth. In the same paper [74], they also proved 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. The NP-completeness proof in [74] implicitly showed that to get C3 we may take $k = 4$ (see also [43]).

We note that C2 also follows for PERFECT MATCHING CUT from a recent result of Bonnet, Chakraborty and Duron [22], who proved that PERFECT MATCHING CUT is NP-complete even for 3-connected subcubic planar graphs. □

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). We give explicit proofs that NODE STEINER TREE and EDGE STEINER TREE are NP-complete on planar subcubic graphs and that this is maintained under subdivision, leading to these two problems being C123-problems.

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 [8] so satisfies C1. For showing C2, we reduce from EDGE STEINER TREE, which is NP-complete even for grid graphs [48], 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 $|E(S)| \leq k$.

Finally, 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'. □

In the Edge Multiway Cut problem, also known as Multiterminal Cut, we are given an input graph $G = (V, E)$, a subset T of its vertices, and an integer k. The goal is to decide whether there exists a set $S \subseteq E$ such that $|S| \leq k$ and for any pair of vertices $\{u, v\} \in T$, $G \backslash S$ does not contain a path between u and v. In the Node Multiway Cut problem, we ask for a set $S \subseteq V \setminus T$ such that $|S| \leq k$ and for any pair of vertices $\{u, v\} \in T$, $G \backslash S$ does not contain a 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 [39] (also following [8]) and NP-complete for planar subcubic graphs [65] 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 [8] (it is an extended monadic second-order linear extremum problem) and NP-complete for planar subcubic graphs [65] so satisfies C1 and C2. It satisfies C3, as we will prove the following claim:

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 [89] and NP-complete for planar subcubic graphs [79] so satisfies C1 and C2. The Induced Disjoint Paths problem is solvable in polynomial time for graphs of bounded mim-width [63] and thus for bounded treewidth [97], 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 [67] for Long Path. The classification of Long Cycle was not made explicit in [6], but is implicitly there (combine Proposition 1 of [6] with Lemma 12 of [6]).

Theorem 18 Long Path, Long Induced Path, Long Cycle *and* Long Induced Cycle *are C123-problems.*

Proof Bodlaender [12] 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 [49], 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 Long Path and Longest Path. Moreover, Long Induced Path satisfies C1; it is even polynomial-time solvable for graphs of bounded mim-width [63]. We can make the same observations for Long Cycle and Long Induced Cycle. □

5 Application to Polynomial-Time Solvable Problems

We give two examples of polynomial-time solvable problems where Theorem 3 gives a distinction between almost-linear-time solvability versus not having a subquadratic-time algorithm (conditional under appropriate hardness hypotheses). Let $d(u, v)$ denote the distance between u and v 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 [42] proved that for subcubic graphs, no subquadratic algorithm exists for DIAMETER under the Orthogonal Vectors Conjecture [99], and no subquadratic algorithm exists for RADIUS under the Hitting Set Conjecture [1]. From the construction in the proof of Evald and Dahlgaard [42], 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 Limitations of our Framework

We give two limitations of our framework.

6.1 Forbidding an Infinite Number of Subgraphs

We observe that in Theorems 1 and 2, the set of graphs $\mathcal{H}$ is allowed to have infinite size. However, the set of graphs $\mathcal{H}$ in Theorems 3 and 6 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. [67], 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 the class of $\mathcal{C}$-subgraph-free graphs (as it satisfies condition C1).

Example 2. Let $\mathcal{H} = \{\mathbb{H}_1, \mathbb{H}_2, \ldots\}$; see also Fig. 2. No graph from $\mathcal{H}$ belongs to $\mathcal{S}$. Every $\mathcal{H}$-subgraph-free graph G is $\mathbb{H}_1$-minor-free. By Theorem 5, G has path-width, and thus treewidth, at most 4. Hence, every C123-problem is efficiently solvable on the class of $\mathcal{H}$-subgraph-free graphs.

6.2 Relaxing Condition C3

In C3 we require the class $\mathcal{G}$ to be subcubic. In this way we are able to show in Theorem 4 that every C123-problem Π satisfies condition D, that is, for every $i \geq 3$, Π is computationally hard for the class of $(C_3, \ldots, C_\ell, K_{1,4}, \mathbb{H}_1, \ldots, \mathbb{H}_\ell)$-subgraph-free graphs.[1] If we allow $\mathcal{G}$ to be any graph class instead of requiring $\mathcal{G}$ to be subcubic, then we can no longer show this, and hence the proof of Theorem 3 no longer holds in that case. 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 $(C_3, \ldots, C_\ell, \mathbb{H}_1, \ldots, \mathbb{H}_\ell)$-subgraph-free. Consequently, in that case, we can only obtain the 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 integers k for which k-subdivision maintains computational hardness is highly problem-specific. For instance, the 1-subdivision of any graph is bipartite and some computationally hard problems, such as Independent Set, become efficiently solvable on bipartite graphs. In the proofs in Sect. 4, k takes on values 1, 2, 3 and 4.

7 Comparison between the Three Frameworks

In this section, we provide an extensive discussion and comparison of the three frameworks in this paper: Theorem 1, 2, and 3. See also Table 1.

7.1 Problems that Belong to all Three Frameworks

Apart from Max- Cut and possibly Tree- Width, all C123-problems from Sect. 4 are NP-complete for planar subcubic graphs, and thus also satisfy the conditions of

[1] The aforementioned papers [46, 82] show a number of problems to be NP-complete for planar subcubic graphs of high girth, whereas we consider subcubic graphs of high girth that, instead of being planar, do not contain any small subdivided "H"-graph as a subgraph.

Theorems 1 and 2. In the proofs of Sect. 4 we made explicit observations about this. The complexity of TREE- WIDTH 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.

7.2 Problems that do not Belong to any of the Three Frameworks

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, [17] for a proof) and thus have tree-width 1.

As another example, the STEINER FOREST problem is to decide for a given integer k, graph G and set of pairs of terminal vertices $S = \{(s_1, t_1), \ldots, (s_p, t_p)\}$, if G has a subforest F with at most k edges, such that s_i and t_i, for every $i \in \{1, \ldots, p\}$, belong to the same connected component of F. It is readily seen that STEINER FOREST generalizes EDGE STEINER TREE: take all pairs of vertices of T as terminal pairs to obtain an equivalent instance of STEINER FOREST. Hence, STEINER FOREST is NP-complete on planar subcubic graphs and this is maintained under subdivision, due to Theorem 15. As STEINER FOREST is NP-complete on graphs of treewidth 3 [10], STEINER FOREST does not belong to any of the three frameworks. We refer to [18] for a partial complexity classification of STEINER FOREST on H-subgraph-free graphs.

As an example on the other extreme end, the CLIQUE problem does not fall under any of the three frameworks for different reasons. As observed in Sect. 1, CLIQUE is polynomial-time solvable for $\mathcal{H}$-subgraph-free graphs for every set of graphs $\mathcal{H}$. Consequently, CLIQUE does not belong to the subgraph framework. Moreover, CLIQUE is polynomial-time solvable for planar graphs, as every clique in a planar graph has size at most 4. Hence, CLIQUE does not belong to the minor and topological minor frameworks either.

7.3 Problems that Only Belong to the Minor Framework

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 its generalization LIST COLOURING [64]). Moreover, 3- COLOURING is NP-complete even for 4-regular planar graphs [34]. Hence, 3- COLOURING belongs to the minor framework. 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 [24].

To give some further examples, we can also take the problems CONNECTED VERTEX COVER, FEEDBACK VERTEX SET and INDEPENDENT FEEDBACK VERTEX SET. It is known that all three problems satisfy C1 [8]. Moreover, CONNECTED VERTEX COVER [48] and FEEDBACK VERTEX SET [92] are NP-complete for planar graphs of maximum degree at most 4. By taking 1-subdivisions, we find that the same holds for Independent Feedback Vertex Set. However, unlike the related problems Vertex Cover and Odd Cycle Transversal, the three problems do not satisfy the conditions of Theorems 2 and 3. This is because Connected Vertex Cover [96], Feedback Vertex Set [96] and Independent Feedback Vertex Set [66] are polynomial-time solvable for subcubic graphs. Munaro [82] showed that even Weighted Feedback Vertex Set is polynomial-time solvable for subcubic graphs.

As a final example, we can take the Matching Cut problem. This problem satisfies C1 [23]. Moreover, it is NP-complete for planar graphs of girth 5 [23] but polynomial-time solvable for subcubic graphs [29].

7.4 Problems that Only Belong to the Minor and Topological Minor Frameworks

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 [9], so satisfies C1, and it is NP-complete for planar subcubic graphs [50] (even if they are also bipartite and have arbitrarily large girth [82]). 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 [75], so it is not a C123-problem. The same holds for Hamilton Path (which contrasts the C123-property of Long Path).

To give another example, Star 3- Colouring is to decide if a graph G has a 3-colouring such that the union of every two colour classes induces a *star forest* (forest in which each connected component is a star). This problem is known to be NP-complete even for subcubic planar subgraphs of arbitrarily large fixed girth [20], but does not satsify C3 [75], so is not C123.

To give a final example of a problem that satisfies the conditions of Theorems 1 and 2 but not those of Theorem 3, we can consider the C_5-Colouring problem. This problem is to decide if a given graph allows a homomorphism to C_5. It is known to be NP-complete on both subcubic graphs [47] and planar graphs [77]. In order to show NP-completeness for subcubic planar graphs, one can take the gadget of MacGillivray and Siggers [77] and augment it with a degree reduction gadget. As explained in Appendix B, where we give a full proof, a suitable gadget appears in the arXiv version of [28].[2] However, C_5-Colouring does not satisfy C3 [75], so it not C123.

7.5 Problems that Only Belong to the Subgraph Framework

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 [60] (and thus also for planar subcubic graphs). However, we show in Sect. 4 that Max- Cut satisfies the conditions of Theorem 3, that is, is a C123-problem.

[2] The use of this gadget for this purpose was proposed to us by Mark Siggers.

8 Conclusions

By giving a meta-classification, we were able to unify a number of known results from the literature, reprove some of them, 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 [7, 55, 67]. 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 [8]. We finish this section by setting out some directions for future work.

8.1 Refining and Extending the Subgraph Framework

We describe three approaches for refining or extending the subgraph framework. First, in the proof of Theorem 4 we gave an example of a C1'D-problem, namely $\mathcal{B}$-Modified List Colouring, that is not C123. However, this example is rather artificial. To increase our understanding of the conditions C1–C3 of our framework, addressing the following question would be helpful.

Open Problem 1 *Do there exist any natural graph C1'D-problems that are not C123-problems?*

As a second approach, we recall from Sect. 6.2 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 2 *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 Sect. 6.1 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 [67], namely for the C123-problem Max- Cut.

Open Problem 3 *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 3, 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, 71, 98] for many involved results in this direction.

8.2 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, which have been frequently stated as open problems before.

Open Problem 4 *Determine the computational complexity of* TREE- WIDTH *for planar graphs and for planar subcubic graphs.*

Open Problem 5 *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 6 *Do there exist other width parameters with the property that the problem of computing them is C123?*

We also made a detailed comparison between the minor, topological minor and subgraph frameworks (see Sect. 7). 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. In particular, we pose the following question.

Open Problem 7 *Does there exist a graph problem that belongs to the minor and subgraph frameworks, but not to the topological minor framework?*

We note that such a problem (if it exists) must be computationally hard for planar graphs and subcubic graphs, but efficiently solvable for subcubic planar graphs.

8.3 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. Recall that we call such problems C23, C13, or C12, respectively.

As discussed in Sect. 1.2, some progress has recently been made on such problems (see e.g. [18, 66, 75]). 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 $\mathbb{H}_i$-subgraph-free graphs (recall that $\mathbb{H}_i$ 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 was not straightforward; recall that Bodlaender et al. [17] essentially settled the computational complexity of Subgraph Isomorphism for H-subgraph-free graphs except if $H = P_5$ or $H = 2P_5$. These cases are open for the minor and topological minor frameworks as well due to the above observation (which also holds for linear forests).

8.4 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. [5, 6, 72, 82]) and treewidth classifications (cf. [2, 3, 71, 98]) 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 6.[3] 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 [76] 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 [76] 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*

[3] The reason is that for any integer k and a sufficiently large integer ℓ, the class of subcubic $(C_3, \ldots, C_\ell, \mathbb{H}_1, \ldots, \mathbb{H}_k)$-free graphs coincides with the class of subcubic $(C_3, \ldots, C_\ell, \mathbb{H}_1, \ldots, \mathbb{H}_k)$-subgraph-free graphs.

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 [62], 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 Weighted Edge Steiner Tree [15], where we allow the edges to have weights. As we showed, even Edge Steiner Tree (the unweighted version) is 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.*

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 *([87]) 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 (or equivalently, if the class of $\mathcal{H}$-minor-free graphs has bounded treewidth) and is computationally hard otherwise.*

Proof 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. Let G be an $\mathcal{H}$-minor-free graph. As G is $\mathcal{H}$-minor-free, G is H-minor-free. We now apply Theorem 21 to find that G has treewidth bounded by some integer c_H, which is a constant as H is a fixed graph. We conclude that the class of $\mathcal{H}$-minor-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.

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. These facts follow directly from results of Robertson and Seymour [86] (see e.g. [21, 32], where this is explained with respect to the more general parameter clique-width). □

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 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 (or equivalently, if the class of* $\mathcal{H}$*-topological-minor-free graphs has bounded treewidth) and is computationally hard otherwise.*

Proof 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. Let G be an $\mathcal{H}$-topological-minor-free graph. As G is $\mathcal{H}$-topological-minor-free, G is H-topological-minor-free. As H is subcubic, this means that G is even H-minor-free. We now apply Theorem 21 to find that G has treewidth bounded by some integer c_H, which is a constant as H 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.

It is well known that for a set of graphs $\mathcal{H}$, a class of $\mathcal{H}$-topological-minor-free graphs has bounded treewidth if and only if $\mathcal{H}$ contains a planar subcubic graph. These facts follow directly from results of Robertson and Seymour [86] (see e.g. [21, 32], where this is explained with respect to the more general parameter clique-width). □

B Hardness of C_5-Colouring for Subcubic Planar Graphs

We show the following result.

Theorem 22 C_5- COLOURING *is* NP*-complete on subcubic planar graphs.*

Proof It is known from [77] that C_5- COLOURING is NP-complete on planar graphs. Let us introduce a degree reduction gadget communicated to us by Mark Siggers (a similar one appears in the proof of Theorem 4.3 from the arXiv version of [28]). The gadget in question is a collection of (some even number) d copies of C_5, joined to one another in sequence by a single overlapping edge, such that the last is joined to the first to form a cycle. The resulting object appears like a flower and is drawn in Fig. 4 for the case $d = 8$. In any homomorphism from this gadget to C_5, the outermost d vertices (i.e., the d vertices of the gadget that have degree 2) must all be mapped to the same vertex of C_5. Thus, from a planar instance G of C_5- COLOURING, we can obtain an equivalent subcubic planar instance G' by replacing in G, each vertex u of degree $d > 3$ with such a gadget in such a way that each of the d neighbours of u is made adjacent to a unique outermost vertex of the gadget. □

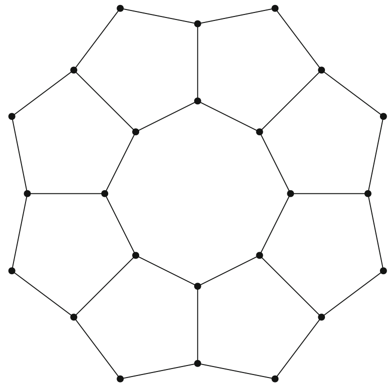

Fig. 4 Degree reduction gadget from Theorem 22 with $d = 8$

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

Author Contributions All seven authors contributed to the paper.

Funding J.J. Oostveen is supported by the NWO grant OCENW.KLEIN.114 (PACAN).

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

Decalarations

Conflict of interest. The authors have no Conflict of interest as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

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**, 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. J. Combin. Theory, Ser. B **164**, 443–472 (2024)
3. Abrishami, T., Chudnovsky, M., Vušković, K.: Induced subgraphs and tree decompositions I. Even-hole-free graphs of bounded degree. J. Combin. Theory Ser. B **157**, 144–175 (2022)
4. Albertson, M.O., Chappell, G.G., Kierstead, H.K., Kündgen, A., Ramamurthi, R.: Coloring with no 2-Colored P_4's. Electron. J. Combin. **11**, R26 (2004)
5. Alekseev, V.E.: On easy and hard hereditary classes of graphs with respect to the independent set problem. Discret. Appl. Math. **132**, 17–26 (2003)

6. Alekseev, V.E., Boliac, R., Korobitsyn, D.V., Lozin, V.V.: NP-hard graph problems and boundary classes of graphs. Theor. Comput. Sci. **389**, 219–236 (2007)
7. Alekseev, V.E., Korobitsyn, D.V.: Complexity of some problems on hereditary graph classes. Diskret. Mat. **4**, 34–40 (1992)
8. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms **12**, 308–340 (1991)
9. Arnborg, S., Proskurowski, A.: Linear time algorithms for NP-hard problems restricted to partial k-trees. Discret. Appl. Math. **23**, 11–24 (1989)
10. Bateni, M., Hajiaghayi, M.T., Marx, D.: Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth. J. ACM **58**(21), 1–21 (2011)
11. Bienstock, D., Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a forest. J. Combin. Theory Ser. B **52**, 274–283 (1991)
12. Bodlaender, H.L.: On linear time minor tests with depth-first search. J. Algorithms **14**, 1–23 (1993)
13. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**, 1305–1317 (1996)
14. Bodlaender, H.L., Bonnet, É., Jaffke, L., Knop, D., Lima, P.T., Milanič, M., Ordyniak, S., Pandey, S., Suchý, O.: Treewidth is NP-complete on cubic graphs. Proc. IPEC 2023, LIPIcs **285**, 7:1–7:13 (2023)
15. Bodlaender, H.L., Brettell, N., Johnson, M., Paesani, G., Paulusma, D., van Leeuwen, E.J.: Steiner trees for hereditary graph classes: A treewidth perspective. Theoret. Comput. Sci. **867**, 30–39 (2021)
16. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) Kernelization. J. ACM **63**(44), 1–44 (2016)
17. 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)
18. Bodlaender, H.L., Johnson, M., Martin, B., Oostveen, J.J., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Complexity framework for forbidden subgraphs IV: The Steiner Forest problem. Proc. IWOCA 2024, LNCS **14764**, 206–217 (2024)
19. Bodlaender, H.L., Kloks, T.: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. J. Algorithms **21**, 358–402 (1996)
20. Bok, J., Jedlicková N., Martin B., Ochem, P., Paulusma D., Smith S.: Acyclic, Star and Injective Colouring: A complexity picture for H-free graphs, CoRR **abs/2008.09415** (full version of Proc. ESA 2020, LIPIcs **173**, 22:1–22:22 (2020))
21. Boliac, R., Lozin, V.V.: On the clique-width of graphs in hereditary classes. Proc. ISAAC 2022, LNCS **2518**, 44–54 (2002)
22. Bonnet, É., Chakraborty, D., Duron, J.: Cutting Barnette graphs perfectly is hard. Proc. WG 2023, LNCS **14093**, 116–129 (2023)
23. Bonsma, P.S.: The complexity of the matching-cut problem for planar graphs and other graph classes. J. Graph Theory **62**, 109–126 (2009)
24. Brook, R.L.: On colouring the nodes of a network. Math. Proc. Cambridge Philos. Soc. **37**, 194–197 (1941)
25. Bulatov, A.A.: A dichotomy theorem for nonuniform CSPs. Proc. FOCS **2017**, 319–330 (2017)
26. 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. Theoret. Comput. Sci. **705**, 75–83 (2018)
27. Chlebík, M., Chlebíková, J.: The complexity of combinatorial optimization problems on d-dimensional boxes. SIAM J. Discret. Math. **21**, 158–169 (2007)
28. Chudnovsky, M., Huang, S., Rzążewski, P.R., Spirkl, S., Zhong M.: Complexity of C_k-coloring in hereditary classes of graphs. Inf. Comput.**292**, 105015 (2023)
29. Chvátal, V.: Recognizing decomposable graphs. J. Graph Theory **8**, 51–53 (1984)
30. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discret. Math. **86**, 165–177 (1990)
31. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Inf. Comput. **85**, 12–75 (1990)
32. Dabrowski, K.K., Paulusma, D.: Clique-width of graph classes defined by two forbidden induced subgraphs. Comput. J. **59**, 650–666 (2016)
33. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**, 864–894 (1994)

34. Dailey, D.P.: Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. Discret. Math. **30**, 289–293 (1980)
35. Darmann, A., Döcker, J., Dorn, B.: The Monotone Satisfiability problem with bounded variable appearances. Int. J. Found. Comput. Sci. **29**, 979–993 (2018)
36. Dawar, A., Grohe, M., Kreutzer, S., Schweikardt, N.: Approximation schemes for first-order definable optimisation problems. Proc. LICS **2006**, 411–420 (2006)
37. Dechter, R., Pearl, J.: Tree clustering for constraint networks. Artif. Intell. **38**, 353–366 (1989)
38. Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. J. ACM **52**, 866–893 (2005)
39. 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)
40. Diestel, R.: Graph Theory, 3rd edition. Springer (2005)
41. Ellis, J.A., Sudborough, I.H., Turner, J.S.: The vertex separation and search number of a graph. Inf. Comput. **113**, 50–79 (1994)
42. Evald, J., Dahlgaard, S.: Tight hardness results for distance and centrality problems in constant degree graphs. CoRR **abs/1609.08403** (2016)
43. Feghali, C., Lucke, F., Paulusma, D., Ries, B.: Matching cuts in graphs of high girth and H-free graphs, Proc. ISAAC 2023, LIPIcs **283**, 31:1–31:16 (2023)
44. Fiorini, S., Hardy, N., Reed, B.A., Vetta, A.: Planar graph bipartization in linear time. Discret. Appl. Math. **156**, 1175–1180 (2008)
45. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Excluded grid minors and efficient polynomial-time approximation schemes. J. ACM **65**(10), 1–10 (2018)
46. Foucaud, F., Hocquard, H., Lajou, D.: Complexity and algorithms for injective edge-coloring in graphs. Inf. Process. Lett. **170**, 106121 (2021)
47. Galluccio, A., Hell, P., Nešetřil, J.: The complexity of H-colouring of bounded degree graphs. Discret. Math. **222**, 101–109 (2000)
48. Garey, M.R., Johnson, D.S.: The Rectilinear Steiner Tree problem is NP complete. J. SIAM Appl. Math. **32**, 826–834 (1977)
49. Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified NP-complete problems. Proc. STOC **1974**, 47–63 (1974)
50. Garey, M.R., Johnson, D.S., Tarjan, R.E.: The Planar Hamiltonian Circuit problem is NP-complete. SIAM J. Comput. **5**, 704–714 (1976)
51. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., USA (1979)
52. Gartland, P., Lokshtanov, D., Masařík, T., Pilipczuk, M., Pilipczuk, M., Rzążewski P.: Maximum Weight Independent Set in graphs with no long claws in quasi-polynomial time. Proc. STOC 2024, 683–691 (2023)
53. Gawrychowski, P., Kaplan, H., Mozes, S., Sharir, M., Weimann, O.: Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. SIAM J. Comput. **50**, 509–554 (2021)
54. Goldreich, O.: Computational complexity: A conceptual perspective. SIGACT News **39**, 35–39 (2008)
55. Golovach, P.A., Paulusma, D.: List coloring in the absence of two subgraphs. Discret. Appl. Math. **166**, 123–130 (2014)
56. Golovach, P.A., Paulusma, D., Ries, B.: Coloring graphs characterized by a forbidden subgraph. Discret. Appl. Math. **180**, 101–110 (2015)
57. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. J. ACM **54**(1), 1–1 (2007)
58. Grohe, M., Marx, D.: Structure theorem and isomorphism test for graphs with excluded topological subgraphs. SIAM J. Comput. **44**, 114–159 (2015)
59. Grzesik, A., Klimosová, T., Pilipczuk, M., Pilipczuk, M.: Polynomial-time algorithm for Maximum Weight Independent set on P_6-free graphs. ACM Trans. Algorithms **18**(4), 1–4 (2022)
60. Hadlock, F.: Finding a maximum cut of a planar graph in polynomial time. SIAM J. Comput. **4**, 221–225 (1975)
61. Harary, F.: Graph theory. Addison-Wesley (1991)
62. Hickingbotham, R.: Induced subgraphs and path decompositions. Electron. J. Combin. **30**, 237 (2023)
63. Jaffke, L., Kwon, O., Telle, J.A.: Mim-width I. induced path problems. Discrete Appl. Math. **278**, 153–168 (2020)

64. Jansen, K., Scheffler, P.: Generalized coloring for tree-like graphs. Discret. Appl. Math. **75**, 135–155 (1997)
65. Johnson, M., Martin, B., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Edge Multiway Cut and Node Multiway Cut are hard for planar subcubic graphs. Proc. SWAT 2024, LIPIcs **294**, 29:1–29:17 (2024)
66. Johnson, M., Martin, B., Pandey, S., Paulusma, D., Smith, S., van Leeuwen, E.J.: Complexity framework for forbidden subgraphs III: When problems are tractable on subcubic graphs. Proc. MFCS 2023, LIPIcs **272**, 57:1-57:15 (2023)
67. Kamiński, M.: Max-Cut and containment relations in graphs. Theoret. Comput. Sci. **438**, 89–95 (2012)
68. Kern, W., Paulusma, D.: Contracting to a longest path in H-free graphs. Proc. ISAAC 2020, LIPIcs **181**, 22:1–22:18 (2020)
69. Kinnersley, N.G.: The vertex separation number of a graph equals its path-width. Inf. Process. Lett. **42**, 345–350 (1992)
70. Kneis, J., Mölle, D., Richter, S., Rossmanith, P.: A bound on the pathwidth of sparse graphs with applications to exact algorithms. SIAM J. Discrete Math. **23**, 407–427 (2009)
71. Korhonen, T.: Grid induced minor theorem for graphs of small degree. J. Combin. Theory Ser. B **160**, 206–214 (2023)
72. Korpelainen, N., Lozin, V.V., Malyshev, D.S., Tiskin, A.: Boundary properties of graphs for algorithmic graph problems. Theoret. Comput. Sci. **412**, 3545–3554 (2011)
73. Kreutzer, S.: Algorithmic meta-theorems. London Math. Soc. Lecture Note Ser. **379**, 177–270 (2011)
74. Le, V.B., Telle, J.A.: The Perfect Matching Cut problem revisited. Proc. WG 2021, LNCS **12911**, 182–194 (2021)
75. Lozin, V.V., Martin, B., Pandey, S., Paulusma, D., Siggers, M., Smith, S., van Leeuwen, E.J.: Complexity framework for forbidden subgraphs II: Edge subdivision and the "H"-graphs. Proc. ISAAC 2024, LIPIcs, to appear
76. Lozin, V.V., Razgon, I.: Tree-width dichotomy. Eur. J. Comb. **103**, 103517 (2022)
77. MacGillivray, G., Siggers, M.H.: On the complexity of H-colouring planar graphs. Discret. Math. **309**, 5729–5738 (2009)
78. Martin, B., Paulusma, D., Smith, S.: Hard problems that quickly become very easy. Inf. Process. Lett. **174**, 106213 (2022)
79. Middendorf, M., Pfeiffer, F.: On the complexity of the disjoint Paths problem. Combinatorica **13**, 97–107 (1993)
80. Mohar, B.: Face covers and the genus problem for apex graphs. J. Combin. Theory, Ser.B **82**, 102–117 (2001)
81. Monien, B., Sudborough, I.H.: Min Cut is NP-complete for edge weighted treees. Theoret. Comput. Sci. **58**, 209–229 (1988)
82. Munaro, A.: Boundary classes for graph problems involving non-local properties. Theoret. Comput. Sci. **692**, 46–71 (2017)
83. Nešetřil, J., de Mendez, P.O.: On nowhere dense graphs. Eur. J. Comb. **32**, 600–617 (2011)
84. Nešetřil, J., de Mendez, P.O.: Sparsity - Graphs, Structures, and Algorithms, Algorithms and combinatorics, vol. 28. Springer (2012)
85. Poljak, S.: A note on stable sets and colorings of graphs. Comment. Math. Univ. Carol. **15**, 307–309 (1974)
86. Robertson, N., Seymour, P.D.: Graph minors. III. Planar tree-width. J. Combin. Theory Ser. B **36**, 49–64 (1984)
87. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. J. Combin. Theory Ser. B **41**, 92–114 (1986)
88. Robertson, N., Seymour, P.D.: Graph minors. XX. Wagner's conjecture. J. Combin. Theory Ser. B **92**, 325–357 (2004)
89. 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)
90. Seese, D.: Linear time computable problems and first-order descriptions. Math. Struct. Comput. Sci. **6**, 505–526 (1996)
91. Siebertz, S.: Nowhere dense graph classes and algorithmic applications. A tutorial at Highlights of Logic, Games and Automata 2019. CoRR **1909.06752** (2019)

92. Speckenmeyer, E.: Untersuchungen zum Feedback Vertex Set Problem in ungerichteten Graphen. Ph.D. thesis, Paderborn (1983)
93. Tamura, Y., Ito, T., Zhou, X.: Algorithms for the independent feedback vertex set problem. IEICE Trans Fundamentals Electron. Commun. Comput. Sci. **98**, 1179–1188 (2015)
94. Telle, J.A., Proskurowski, A.: Algorithms for vertex partitioning problems on partial k-trees. SIAM J. Discret. Math. **10**, 529–550 (1997)
95. Telle, J.A., Villanger, Y.: FPT algorithms for domination in sparse graphs and beyond. Theoret. Comput. Sci. **770**, 62–68 (2019)
96. Ueno, S., Kajitani, Y., Gotoh, S.: On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. Discret. Math. **72**, 355–360 (1988)
97. Vatshelle, M.: New Width Parameters of Graphs. Ph.D. thesis, University of Bergen (2012)
98. Weißauer, D.: In absence of long chordless cycles, large tree-width becomes a local phenomenon. J. Combin. Theory, Ser. B **139**, 342–352 (2019)
99. Williams, R.: A new algorithm for optimal 2-constraint satisfaction and its implications. Theoret. Comput. Sci. **348**, 357–365 (2005)
100. Williams, V.V.: On some fine-grained questions in algorithms and complexity. Proc. ICM **2018**, 3447–3487 (2018)
101. Wolle, T.: Computational Aspects of Treewidth: Lower Bounds and Network Reliability, Ph.D. thesis, Utrecht University (2005)
102. Xi, W., Lin, W.: On maximum P_3-packing in claw-free subcubic graphs. J. Comb. Optim. **41**, 694–709 (2021)
103. Yannakakis, M.: Node and edge deletion NP-complete problems. Proc. STOC **1978**, 253–264 (1978)
104. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. SIAM J. Appl. Math. **38**, 364–372 (1980)
105. Zhuk, D.: A proof of the CSP dichotomy conjecture. Journal of the ACM **67**, 30:1–30:78 (2020)