arXiv:1309.1559v1 [cs.DS] 6 Sep 2013

Large induced subgraphs via triangulations and CMSO

Fedor V. Fomin[1], Ioan Todinca[*] [2], and Yngve Villanger[1]

[1]Department of Informatics, University of Bergen, Norway, fomin@ii.uib.no, yngve.villanger@uib.no

[2]LIFO, Univ. Orléans, France, ioan.todinca@univ-orleans.fr

November 10, 2018

Abstract

We obtain an algorithmic meta-theorem for the following optimization problem. Let φ be a Counting Monadic Second Order Logic (CMSO) formula and $t \geq 0$ be an integer. For a given graph $G = (V, E)$, the task is to maximize $|X|$ subject to the following: there is a set $F \subseteq V$ such that $X \subseteq F$, the subgraph $G[F]$ induced by F is of treewidth at most t, and structure $(G[F], X)$ models φ, i.e. $(G[F], X) \models \varphi$. Special cases of this optimization problem are the following generic examples. Each of these special cases contains various problems as a special subcase:

- Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles, where for fixed nonnegative integers m and ℓ, the task is to find a maximum induced subgraph of a given graph with at most ℓ vertex-disjoint cycles of length $0 \pmod{m}$. For example, this encompasses the problems of finding a maximum induced forest or a maximum subgraph without even cycles.
- Minimum $\mathcal{F}$-Deletion, where for a fixed finite set of graphs $\mathcal{F}$ containing a planar graph, the task is to find a maximum induced subgraph of a given graph containing no graph from $\mathcal{F}$ as a minor. Examples of Minimum $\mathcal{F}$-Deletion are the problems of finding a minimum vertex cover or a minimum number of vertices required to delete from the graph to obtain an outerplanar graph.
- Independent $\mathcal{H}$-packing, where for a fixed finite set of connected graphs $\mathcal{H}$, the task is to find an induced subgraph F of a given graph with the maximum number of connected components, such that each connected component of F is isomorphic to some graph from $\mathcal{H}$. For example, the problem of finding a maximum induced matching or packing into nonadjacent triangles, are the special cases of this problem.

We give an algorithm solving the optimization problem on an n-vertex graph G in time $\mathcal{O}(|\Pi_G| \cdot n^{t+4} \cdot f(t, \varphi))$, where Π_G is the set of all potential maximal cliques in G and f is a function of t and φ only. We also show how similar running time can be obtained for the weighted version of the problem. Pipelined with known bounds on the number of potential maximal cliques, we derive a plethora of algorithmic consequences extending and subsuming many known results on algorithms for special graph classes and exact exponential algorithms.

[*]Partially supported by the ANR project AGAPE.

1 Introduction

We provide a generic algorithmic result concerning induced subgraphs with properties expressible in some logic. The main applications of our result can be found in two areas of graph algorithms: polynomial time algorithms on special graph classes and exponential time algorithms.

Graph classes. The algorithmic study of graphs with particular structure can be traced to the introduction of perfect graphs by Berge in the beginning of 1960s. Most of the research in this area focuses on graph algorithms exploiting the structure of the input graph. Many problems intractable on general graphs were shown to be solvable in polynomial time on different classes of graphs like interval or chordal graphs. The book of Golumbic [44] provides algorithmic studies of fundamental classes of perfect graphs while the book of Brandstädt et al. [15] gives an extensive overview of different classes of graphs. By the seminal work of Grötschel et al. [47], the weighted versions of Maximum Independent Set, Maximum Clique, Coloring, and Minimum Clique Cover are solvable in polynomial time on perfect graphs. There are two natural research directions in this area extending the limits of tractability. One direction is to identify graph classes beyond perfect graphs, where a specific problem like Maximum Independent Set, can still be solved efficiently. The second direction is to identify more general problems which still can be solved in polynomial time on subclasses of perfect graphs.

As an example, let us take Maximum Induced Forest[1], which can be seen as a natural extension of Maximum Independent Set, where instead of maximum edgeless graph one is seeking for a maximal acyclic graph. It easy to notice that the problem is NP-complete being restricted to bipartite, and thus to perfect, graphs. On the other hand, for other classes of graphs the problem is solvable in polynomial time. Yannakakis and Gavril [73] have shown how to find in polynomial time a maximum induced forest and tree on chordal graphs. In fact, they show polynomial time solvability of more general problem of finding maximum and connected maximum k-colorable subgraphs in chordal graphs, where k is a constant. When k is a part of the input, they showed that on chordal graphs both problems are NP-compete. Other graph classes where Maximum Induced Forest was known to be solvable in polynomial time include circle n-gon graphs, circle trapezoid, circle graphs, and bipartite chordal graphs [41, 42, 53]. The containment relations between these classes of graphs is given in Fig 1. According to the database `http://www.graphclasses.org` on special graph classes the complexity of (weighted) Maximum Induced Forest on weakly chordal is open.

Another example of a well-studied problem on special graph classes is Maximum Induced Matching. Here the task is to find a maximum induced subgraph such that every connected component of this graph is an edge. The complexity of this problem on different graph classes was investigated in [17, 19, 20, 45]. Cameron and Hell in [18] introduced the following generalization of Maximum Induced Matching. Let $\mathcal{H}$ be a finite set of connected graphs. An $\mathcal{H}$-packing of a given graph G is a pairwise vertex-disjoint set of subgraphs of G, each isomorphic to a member of $\mathcal{H}$. An independent $\mathcal{H}$-packing of a given graph G is an $\mathcal{H}$-packing, i.e. a set of pairwise vertex-disjoint set of subgraphs of G, each isomorphic to a member of $\mathcal{H}$, such that no two subgraphs of the packing are joined by an edge of G. The task is to find the

[1] In the literature, the complementary minimization problem of deleting the minimum number of vertices such that the remaining graphs has no cycles, is known as Minimum Feedback Vertex Set. Since from exact algorithms perspective maximization and minimization versions are equivalent, we will be discussing mostly maximization problems.

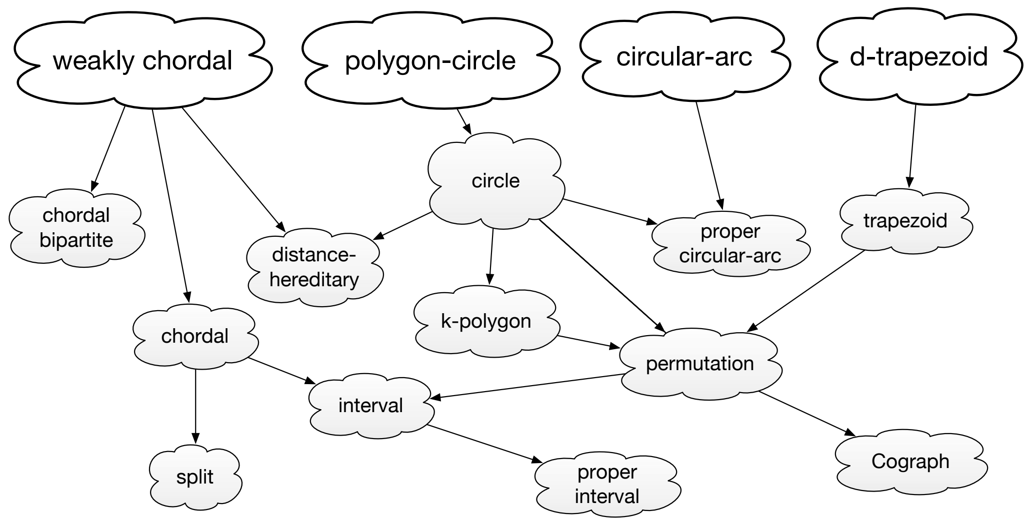

Figure 1: Graph classes with a polynomial number of potential maximal cliques.

maximum number of graphs contained in an independent $\mathcal{H}$-packing. For example, when $\mathcal{H}$ consists of K_1 this is Maximum Independent Set, and when $\mathcal{H} = \{K_2\}$, this is Maximum Induced Matching. It has been shown in [18] that for many graph classes including weakly chordal and polygon-circle graphs, $\mathcal{H}$-packing is solvable in polynomial time.

Exact exponential algorithms. The second application of our results can be found in the area of exact exponential algorithms. The area of exact exponential algorithms is about solving intractable problems faster than the trivial exhaustive search, though still in exponential time [31]. While for any graph property π testable in polynomial time, the problem of finding a maximum induced subgraph with property π is trivially solvable in time $2^n n^{\mathcal{O}(1)}$, for several fundamental problems much faster algorithms are known. A longstanding open question in the area is if Maximum Induced Subgraph with Property π can be solved faster than the trivial $\mathcal{O}^*(2^n)$ for every hereditary property π testable in polynomial time.

For the simplest property π, being edge-less, the corresponding maximum induced subgraph problem is Maximum Independent Set. A significant amount of research was also devoted to algorithms for this problem starting from the classical work of Moon and Moser [59] (see also Miller and Muller [58]) from the 1960s [69, 50, 66, 30, 14, 54]. To the best of our knowledge, the fastest known algorithm of running time $\mathcal{O}(1.2109^n)$ is due to Robson [66]. For Maximum Induced Forest an algorithm of running time $\mathcal{O}(1.7548^n)$ was known [28]. This result was improved and generalized by a subset of the authors, who have shown that for any fixed t, the maximum induced subgraph of treewidth at most t can be computed in time $\mathcal{O}(1.7347^n)$ [35]. There is also a relevant work of Gupta et al. [48] who gave algorithms for Maximum Induced Matching and Maximum 2-Regular Induced Subgraph, with running times time $\mathcal{O}(1.695733^n)$ and $\mathcal{O}(1.7069^n)$, respectively.

Our main theorem is based on developments from two research areas: the theory of minimal triangulations and logic.

Minimal triangulations. A triangulation of a graph G is a chordal (no induced cycle of length at least four) supergraph of G. A triangulation H of G is minimal, if no proper subgraph of H is a triangulation of G. Triangulations are closely related to fundamental problems arising in sparse matrix computations which were studied intensively in the past [60, 67]. The survey of Heggernes [49] gives an overview of techniques and applications of minimal triangulations. It appeared in 1990s that minimal separators play important role in obtaining minimal triangulations with certain properties. Techniques based on minimal separators were used to obtain polynomial algorithm computing the treewidth and minimum fill-in for different classes of graphs [10, 52, 51]. These results were extended by Bouchitté and Todinca in [12, 13], who also introduced the notion of a potential maximal clique, which is a set of vertices of a graph that is a clique in some minimal triangulation. Potential maximal cliques appeared to be a handy tool for computing the treewidth of a graph [32, 37]. Recently potential maximal clique based machinery was used to obtain a subexponential parameterized algorithm finding a minimum fill-in of a graph [36]. The work which is most relevant to our results is the work of a subset of the authors [35], where potential maximal cliques were used to find maximum induced subgraphs of treewidth at most t. We build on the previous techniques exploiting the structure of minimal triangulations, minimal separators and potential maximal cliques but to use the framework of minimal triangulations in full generality, we have to combine it with the powerful tools from logic.

Algorithmic applications of logic. Algorithmic meta-theorems are algorithmic results which can be applied to large families of combinatorial problems, instead of just specific problems. Such theorems provide a better understanding of the scope of general algorithmic techniques and the limits of tractability. Usually meta-theorems are based on the deep relations between logic and combinatorial structures, which is a fundamental issue of computational complexity [46, 56]. A typical example of a meta-theorem is the celebrated Courcelle's theorem [23] which states that all graph properties definable in Monadic Second Order Logic can be decided in linear time on graphs of bounded treewidth. More recent examples of such meta-theorems state that all first-order definable properties on planar graphs can be decided in linear time [38], that all first-order definable optimization problems on classes of graphs with excluded minors can be approximated in polynomial time to any given approximation ratio [26], and that all parameterized problems with finite integer index and additional "compactness" or "bidimensional" combinatorial property, admit linear kernels on planar graphs [9, 34]. As it often happens with meta-theorems, a combination of logic and graph theory not only give a uniform explanation to tractability of many algorithmic problems but also provide a variety of new results. There are several extensions of Courcelle's theorem known in the literature. In particular, for a counting variant of MSO, Counting Monadic Second Order Logic (CMSO), where we are allowed to have sentences testing if a set is equal to q modulo r, for some integers q and r, and analogue of Courcelle's theorem was obtained by Borie et al. [11] and Lagergren and Arnborg [57]. Our proof is using the framework of Borie et al. [11].

Our results. A *property* $\mathcal{P}(G, X)$ on graphs, where G is a graph and X is a vertex subset of its vertices, associates to each graph G and each vertex subset X of G a boolean value. Borie et al. [11] defined *regular properties*, which definition we postpone till the next section. For all our applications, we need only the fact from Borie et al. [11] that every property $\mathcal{P}(G, X)$ expressible by a CMSO-formula is regular. Then our result can be stated as follows. Let φ be a CMSO-formula, $G = (V, E)$ be a graph, and $t \geq 0$ be an integer. We consider the following optimization problem

$$\begin{array}{ll} \text{Max} & |X| \\ \text{subject to} & \text{There is a set } F \subseteq V \text{ such that } X \subseteq F; \\ & \text{The treewidth of } G[F] \text{ is at most } t; \\ & (G[F], X) \models \varphi. \end{array} \tag{1}$$

For example, Maximum Independent Set can be encoded by (1) by taking $t = 0$, and φ expressing that $X = F$ and the absence of edges in $G[F]$. For another example, consider Independent Cycle Packing, where the task is to find an induced subgraph with maximum number of connected components such that each component is a cycle. In this case, $t = 2$ and φ expresses the property that each connected component is a cycle and that X is a set of vertices containing exactly one vertex from each cycle.

Let Π_G be the set of all potential maximal cliques in G. Our main result is that (1) is solvable in time $O(|\Pi_G| \cdot |V|^{t+4} \cdot f(t, \varphi))$ for some function f. Moreover, within the same running time one can find the corresponding sets X and F. Also it is easy to extend our algorithm to solve within the same running time weighted and annotated versions of (1).

Many well studied graph classes have the following property: there is a polynomial function p, depending only on the graph class, such that for every graph G from the class, the number of potential maximal cliques in G is at most $p(n)$, see Fig 1 for examples of such classes. Moreover, if the number of potential maximal cliques in a graph is bounded by some polynomial of n, then all potential maximal cliques can be enumerated in polynomial time [13]. Our algorithm implies directly that every problem expressible in the form of (1) is solvable in polynomial time on such graph classes. We discuss in details the bounds on the number of potential maximal cliques for different graph classes in Section 5. Interestingly enough, while recognition of several of graph classes, like polygon-circle or d-trapezoid, can be NP-complete, our algorithm is still able either to solve the problem, or to report that the input graph does not belong to the specified graph class. Such algorithms were called *robust* by Raghavan and Spinrad [62]. To the best of our knowledge, very few robust algorithms were known in the literature prior to our work.

Another direct consequence of our algorithm is that because every n-vertex graph has $\mathcal{O}(1.7347^n)$ potential maximal cliques [35], many intractable problems concerning maximum induced subgraphs with different properties expressible in the form of (1), can be solved significantly faster than by the trivial $\mathcal{O}(2^n)$-time brute-force algorithm. We are not aware of any algorithmic meta-result of this flavor in the area of exact algorithms.

We mention below the most interesting special cases of the optimization problem (1). Each of these special cases contains various problems as a special subcase, we discuss subcases after introducing each of the problems. For some of these cases, expressibility in the form of (1) is trivial but for some it is non-obvious and requires deep results from Graph Theory. We discuss these issues in more details in Section 4.

Let $\mathcal{F}_m$ be the set of cycles of length $0 \pmod{m}$. Let $\ell \geq 0$ be an integer. Our first example is the following problem.

Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles
Input: A graph G.
Task: Find a set $F \subseteq V(G)$ of maximum size such that $G[F]$ contains at most ℓ vertex-disjoint cycles from $\mathcal{F}_m$.

Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles encompasses several interesting problems. For example, when $\ell = 0$, the problem is to find a maximum induced subgraph without cycles divisible by m. For $\ell = 0$ and $m = 1$ this is Maximum Induced Forest.

For integers $\ell \geq 0$ and $p \geq 3$, the problem related to Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles is the following.

Maximum Induced Subgraph with $\leq \ell$ copies of p-cycles
Input: A graph G.
Task: Find a set $F \subseteq V(G)$ of maximum size such that $G[F]$ contains at most ℓ vertex-disjoint cycles of length at least p.

Next example concerns properties described by forbidden minors. Graph H is a *minor* of graph G if H can be obtained from a subgraph of G by a (possibly empty) sequence of edge contractions. A *model* M of minor H in G is a minimal subgraph of G, where the edge set $E(M)$ is partitioned into *c-edges (contraction edges)* and *m-edges (minor edges)* such that the graph resulting from contracting all c-edges is isomorphic to H. Thus, H is isomorphic to a minor of G if and only if there exists a model of H in G. For an integer ℓ a finite set of graphs $\mathcal{F}$, we define he following generic problem.

Maximum Induced Subgraph with $\leq \ell$ copies of Minor Models from $\mathcal{F}$
Input: A graph G.
Task: Find a set $F \subseteq V(G)$ of maximum size such that $G[F]$ contains at most ℓ vertex disjoint minor models of graphs from $\mathcal{F}$.

Even the special case with $\ell = 0$, this problem and its complementary version called the Minimum $\mathcal{F}$-Deletion, encompass many different problems. In the literature, the case $\ell = 0$ was studied from parameterized and approximation perspective [33].

When $\mathcal{F} = \{K_2\}$, a complete graph on two vertices, this is Maximum Independent Set, the problem complementary to the Minimum Vertex Cover problem. When $\mathcal{F} = \{C_3\}$, a cycle on three vertices, this is Maximum Induced Forest. Case $\mathcal{F} = \{K_4\}$ of Maximum Induced $\mathcal{F}$-free Subgraph corresponds to maximum induced serial-parallel graph, $\mathcal{F} = \{K_4, K_{2,3}\}$ to maximum induced outerplanar, and case when $\mathcal{F}$ consists of a diamond graph, which is K_4 minus one edge, is to find a maximum induced cactus subgraph. Maximum induced pseudo-forest is the case of $\mathcal{F}$ containing the diamond and butterfly graphs, which is obtained by identifying one vertex of two triangles. Maximum Apollonian graph corresponds to the situation with $\mathcal{F}$ consisting of the complete graph K_5, the complete bipartite graph $K_{3,3}$, the graph of the octahedron, and the graph of the pentagonal prism. A fundamental problem, which is a special case of Minimum $\mathcal{F}$-Deletion, is Minimum Treewidth η-Deletion or η-Transversal which is to delete minimum vertices to obtain a graph of treewidth at most η. Since by the result of Robertson and Seymour [63] every graph of treewidth η excludes a $(\eta+1) \times (\eta+1)$ grid as a minor, we have that the set $\mathcal{F}$ of forbidden minors of treewidth η graphs contains a planar graph. Similarly, for $\ell > 0$, Maximum Induced Subgraph with $\leq \ell$ copies of Minor Models from $\mathcal{F}$ generalizes problems like finding a maximum induced subgraph containing at most ℓ vertex-disjoint cycles, at most ℓ vertex-disjoint outerplanar graphs, at most ℓ vertex-disjoint subgraphs of treewidth t, etc. For some graph classes, like circular-arc and weakly chordal, we show that even more general cases of Minimum $\mathcal{F}$-Deletion, when $\mathcal{F}$ is not requested to contain a planar graph, are still solvable in polynomial time.

Let $t \geq 0$ be an integer and φ be a CMSO-formula. Let $\mathcal{G}(t,\varphi)$ be a class of connected graphs of treewidth at most t and with property expressible by φ. Our next example is the following problem.

INDEPENDENT $\mathcal{G}(t,\varphi)$-PACKING
Input: A graph G.
Task: Find a set $F \subseteq V(G)$ with maximum number of connected components such that each connected component of $G[F]$ is in $\mathcal{G}(t,\varphi)$.

In other words, the task is to find a maximum vertex-disjoint packing in G of subgraphs from $\mathcal{G}(t,\varphi)$ such that no two subgraphs of the packing are joined by an edge of G. This problem trivially generalizes several well studied problems. For example, MAXIMUM INDUCED MATCHING is to find a maximum induced matching which was studied intensively for different graph classes. Similarly, when class $\mathcal{G}(t,\varphi)$ consists of one graph K_3, then MAXIMUM INDUCED $\mathcal{G}(t,\mathcal{P})$-PACKING is induced triangle packing. This problem, under the name INDEPENDENT TRIANGLE PACKING was studied by Cameron and Hell [18]. Recall that Cameron and Hell defined more general problem, namely, INDEPENDENT $\mathcal{H}$-PACKING, where for a finite set of connected graphs $\mathcal{H}$, the task is to find a maximum number of disjoint copies of graphs from $\mathcal{H}$ such that there is no edges between the copies. Since every finite set of graphs is trivially in $\mathcal{G}(t,\mathcal{P})$ for some t and $\mathcal{P}$, INDEPENDENT $\mathcal{H}$-PACKING is a special case of INDEPENDENT $\mathcal{G}(t,\varphi)$-PACKING. Another studied variant of the problem is INDUCED PACKING OF ODD CYCLES introduced by Golovach et al. in [43], where the task is to find the maximum number of odd cycles such that there is no edge between any pair of cycles.

The next problem is an example of annotated version of optimization problem (1).

k-IN-A-GRAPH FROM $\mathcal{G}(t,\varphi)$
Input: A graph G, with k terminal vertices.
Task: Find an induced graph from $\mathcal{G}(t,\varphi)$ containing all k terminal vertices.

It is also easy to handle variants of this problem where terminal vertices have specific properties, like being the endpoints of the path if $\mathcal{G}(t,\varphi)$ is the class of paths. Many variants of k-IN-A-GRAPH FROM $\mathcal{G}(t,\varphi)$ can be found in the literature, like k-IN-A-PATH, k-IN-A-TREE, k-IN-A-CYCLE. k-IN-A-PATH is clearly solvable in polynomial time for $k = 2$. For $k = 3$ the problem is NP-complete already on graph of maximum vertex degree at most three [27]. Bienstock [6] have shown that the following cases of k-IN-A-GRAPH FROM $\mathcal{G}(t,\varphi)$ are NP-hard: finding an induced odd cycle of length greater than three, passing through a prescribed vertex and finding an induced odd path between two prescribed vertices. Polynomial time algorithms for the odd path problem are known for several graph classes, including chordal [1] and circular-arc graphs [2]. Chudnovsky and Seymour have shown that k-IN-A-TREE for $k = 3$ is solvable in polynomial time [21]. The complexity of the case $k = 4$ is open.

Let us remark that because of the power of CMSO, different modifications of the problems mentioned above, with additional requirements on the induced subgraph like being connected, constrains on vertex degree and parities of connected components, can be easily handled.

2 Preliminaries

We denote by $G = (V, E)$ a finite, undirected and simple graph with $|V| = n$ vertices and $|E| = m$ edges. Sometimes the vertex set of a graph G is referred to as $V(G)$ and its edge set as $E(G)$. A clique K in G is a set of pairwise adjacent vertices of $V(G)$. The *neighborhood* of a vertex v is $N(v) = \{u \in V : \{u, v\} \in E\}$. For a vertex set $S \subseteq V$ we denote by $N(S)$ the set $\bigcup_{v \in S} N(v) \setminus S$.

The notion of treewidth is due to Robertson and Seymour [63]. A *tree decomposition* of a graph $G = (V, E)$, denoted by $TD(G)$, is a pair (X, T), where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a family of subsets of V, called *bags*, such that

(i) $\bigcup_{i \in V(T)} X_i = V$,

(ii) for each edge $e = \{u, v\} \in E(G)$ there exists $i \in V(T)$ such that both u and v are in X_i, and

(iii) for all $v \in V$, the set of nodes $\{i \in V(T) \mid v \in X_i\}$ induces a connected subtree of T.

The maximum of $|X_i| - 1$, $i \in V(T)$, is called the *width* of the tree decomposition. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width taken over all tree decompositions of G.

Counting Monadic Second Order Logic. We use Counting Monadic Second Order Logic (CMSO), an extension of MSO, as a basic tool to express properties of vertex/edge sets in graphs.

The syntax of Monadic Second Order Logic (MSO) of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges, sets of vertices, and sets of edges, the quantifiers $\forall$, $\exists$ that can be applied to these variables, and the following five binary relations:

1. $u \in U$ where u is a vertex variable and U is a vertex set variable;
2. $d \in D$ where d is an edge variable and D is an edge set variable;
3. $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident with the vertex u;
4. $\mathbf{adj}(u, v)$, where u and v are vertex variables and the interpretation is that u and v are adjacent;
5. equality of variables representing vertices, edges, sets of vertices, and sets of edges.

In addition to the usual features of monadic second-order logic, if we have atomic sentences testing whether the cardinality of a set is equal to q modulo r, where q and r are integers such that $0 \leq q < r$ and $r \geq 2$, then this extension of the MSO is called the *counting monadic second-order logic*. So essentially CMSO is MSO with the following atomic sentence for a set S:

$$\mathbf{card}_{q,r}(S) = \mathbf{true} \text{ if and only if } |S| \equiv q \pmod{r}.$$

We refer to [3, 22, 24] and the book of Courcelle and Engelfriet [25] for a detailed introduction on CMSO. In [25], the CMSO is referred to as CMS_2.

2.1 Treewidth, t-terminal recursive graphs and regular properties

We use one of the (many) alternative definitions of treewidth, based on *terminal graphs*. A *t-terminal graph* $G = (V, T, E)$ is a graph with an ordered set $T \subseteq V$ of at most t distinguished vertices, called *terminals*. Denote by $\tau(G)$ the number of terminals of graph G.

A t-terminal graph (V, T, E) is a *base graph* if $V = T$. We define *composition operations* over the set of t-terminal graphs. A composition operation f is of arity 1 or 2. When f is of arity 2, it acts on two t-terminal graphs G_1, G_2 and produces a t-terminal graph $G = f(G_1, G_2)$ as follows. It first makes disjoint copies of the two graphs, then "glues" some terminals of graphs G_1 and G_2. Operation f is represented by a matrix $m(f)$. The matrix has 2 columns and $\tau(G) \leq t$ lines, its values are integers between 0 and t. At line i of the matrix, elements $m_{ij}(f)$ indicate which terminals of graphs G_j are identified to terminal number i of G. If $m_{ij}(f) = 0$ it means that no terminal of G_j was identified to terminal number i of G. A terminal of G_j can be identified to at most one terminal of G (a column j cannot contain two equal, non-zero values). Note that if $m_{i1}(f) = 0$ and $m_{i2}(f) = 0$ it means that terminal i of G is a new vertex.

When f is of arity 1, its matrix $m(f)$ has only one column. The t-terminal graph $G = f(G_1)$ is obtained from graph G_1 and matrix $m(f)$ as above, by identifying terminal $m_{i1}(f)$ to terminal number i in G.

Observe that the number of possible composition operations over t-terminal graphs is bounded by some function of t. We say that a t-terminal graph G is *t-terminal recursive* if it can be obtained from t-terminal base graphs through a sequence of composition operations. This sequence is called the *t-expression* of graph G.

Proposition 1 ([8])**.** *For any $(t+1)$-terminal recursive graph $H = (V, T, E)$, there is a tree decomposition of (V, E) of width at most t, with a bag containing T. Conversely, for any tree decomposition of width t of graph $G = (V, E)$ and any bag W of the decomposition, (V, W, E) is a $(t+1)$-terminal recursive graph.*

Proof. Assume that (V, T, E) can be obtained recursively, through composition operations, from $(t+1)$-terminal base graphs. The expression constructing this graph can be represented as a tree, the leaves being the base graphs, each internal node corresponding to a composition operation. The tree decomposition of G is simply obtained by following this tree and putting, in each node, a bag corresponding to the terminals of the graph represented by the corresponding sub-expression. The bags are clearly of size at most $t+1$. One can easily check that the set of bags satisfies the conditions of a tree decomposition.

The other direction is proved in [8], Theorem 40. □

Consider a *property* $\mathcal{P}(G, X)$ on graphs depending on a vertex subset X. That is, property $\mathcal{P}$ associates to each graph G and each vertex subset X of G a boolean value. By the celebrated results of [22, 3, 11], it is well-known that if the property can be expressed by a CMSO-formula, there exists a linear-time algorithm taking as input a $(t+1)$-terminal recursive graph $G = (V, T, E)$ and computing a maximum (or minimum) size vertex set X such that $\mathcal{P}(G, X)$. Many natural problems like Maximum Independent Set or Minimum Dominating Set can be expressed in this setting.

Typical algorithms for such problems proceed by dynamic programming. When browsing the $(t+1)$-expression of G, the algorithm stores in each node a table of *classes* (sometimes called *characteristics*) depending on the branch of the current sub-expression and the partial solutions (i.e., possible subsets of X) encountered so far. Let G_1 be such a sub-expression and let X_1 be a subset of vertices that we aim to extend into the solution X. The intuition is that if the class of (G_1, X_1) is the same as the class of some other pair (G_2, X_2), then we can replace the branch of G_1 by an expression of G_2, and the new graph G' is such that X_1 extends into a solution $X_1 \cup Y$ of G if and only if X_2 extends into a solution $X_2 \cup Y$ of G'.

In order to efficiently solve our problem, we need an efficient computation of classes for base graphs, as well as an efficient computation of the classes for compositions of graphs and partial solutions.

We give a formal definition of these "good" properties; the vocabulary is inspired by Borie *et al.* [11].

Let now $G = (V, T, E)$ be a $(t+1)$-terminal recursive graph. For any composition operation f, let $\circ_f$ denote the composition operation over pairs (G, X), where f extends in a natural way over the values of vertex sets. If $G = f(G_1)$ then $\circ_f((G_1, X)) = (G, X)$. If $G = f(G_1, G_2)$ then $\circ_f((G_1, X_1), (G_2, X_2)) = (G, X)$, the operation being valid only if, for each terminal of G, either we have mapped terminals from both G_1 and G_2, contained in both X_1 and X_2, or we have not mapped any terminal belonging to X_1 or X_2. Then X is obtained from X_1 and X_2 by merging those vertices corresponding to terminals that have been mapped on a same terminal of G.

Definition 1 (Regular Property)**.** *Consider a property $\mathcal{P}(G, X)$ over graphs and corresponding vertex subsets. Property $\mathcal{P}$ is called* regular *if, for every t, there exists a finite set $\mathcal{C}$, a homomorphism h associating to each $(t+1)$-terminal recursive graph G and every $X \subseteq V(G)$ a class $h(G, X) \in \mathcal{C}$, and an* update function *$\odot_f : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ for each composition operation f of arity 2 (resp. $\odot_f : \mathcal{C} \to \mathcal{C}$ for each composition operation f of arity 1), satisfying:*

- *(property $\mathcal{P}$ is preserved) If $h(G_1, X_1) = h(G_2, X_2)$ then $\mathcal{P}(G_1, X_1) = \mathcal{P}(G_2, X_2)$.*
- *(integrity of operations) For any composition operation f, we have that*

$$h(\circ_f((G_1, X_1), (G_2, X_2))) = \odot_f(h(G_1, X_1), h(G_2, X_2))$$

if f is of arity 2, and

$$h(\circ_f(G_1, X_1)) = \odot_f(h(G_1, X_1))$$

if f is of arity 1.

We point out that the homomorphism class $h(G, X)$ depends on G and on the value of X. Typically the class of $h(G, X)$ encodes, among other informations, the intersection of X with the set of terminals. For example, if the composition operation $\circ_f((G_1, X_1), (G_2, X_2))$ is not valid, then $\odot_f(c_1, c_2)$, where c_1 and c_2 are the respective homomorphism classes of (G_1, X_1) and of (G_2, X_2), is also undefined.

Note that for any fixed t and any regular property $\mathcal{P}$, the number of classes is constant. Nevertheless, this constant depends on t and on the property $\mathcal{P}$. For algorithmic purposes, given t and $\mathcal{P}$, we need an explicit algorithm computing the homomorphism class of a given base graph, and an algorithm computing the update functions $\odot_f$. I.e., we need an algorithm that takes as input a composition operation f and one or two classes $c_1, c_2 \in \mathcal{C}$ and computes the class $\odot_f(c_1, c_2)$ if f is of arity 2 (resp. $\odot_f(c_1)$ if f is of arity 1). Eventually, we must know the set of *accepting classes*, that is the set of classes c such that $h(G, X) = c$ implies that $\mathcal{P}(G, X)$.

As an example, consider the property $3COL(G, X)$ which is true only if $G[X]$ is 3-colourable. We show that it is regular. Let $P_3(t)$ be the set of partitions of subsets of $\{1, 2 \dots, t+1\}$ into three parts. The set $\mathcal{C}$ of homomorphism classes is $P_3(t)$. Consider a $(t+1)$-terminal recursive graph $G = (V, T, E)$ and let $X \subseteq V$. For each 3-partition (X_1, X_2, X_3) of the vertex subset X into three independent sets, let $p(X_1, X_2, X_3) \in P_3(t)$ be the 3-partition of $T \cap X$ corresponding to $(T \cap X_1, T \cap X_2, T \cap X_3)$; here, for $T \cap X_i$, we only keep the ranks of the terminals of $T \cap X_i$ in the ordered set T. The class $h(G, X)$ will be $\{p(X_1, X_2, X_3) \mid (X_1, X_2, X_3)$ is a partition of V into three independent sets$\}$. In particular, the unique non-accepting class is $\emptyset$. It is not hard to see that, for fixed t, the class of every base graph can be computed in constant time, and that for any composition operation f the update function $\odot_f$ exists and can also be computed in constant time. The number of classes is constant even though the number of subsets X is arbitrarily large. When solving the problem $\max |X| : 3COL(G, X)$ on a $(t+1)$-terminal recursive graph G, we must store, in each node u of the $(t+1)$-expression, for each class c, the size of the maximum vertex subset X_u of the current graph G_u such that $h(G_u, X_u) = c$. The overall solution is the maximum one among the accepting classes of the root node.

We say that a CMSO-formula φ *expresses* a property $\mathcal{P}(G, X)$ if $\mathcal{P}(G, X)$ is true if and only if (G, X) models φ (i.e., the formula is true exactly on graphs G and vertex subsets X such that $\mathcal{P}(G, X)$ is true).

Proposition 2 (Borie *et al.* [11])**.** *Any property $\mathcal{P}(G, X)$ expressible by a CMSO-formula is regular.*

Moreover, the result of Borie *et al.* [11] is constructive in the sense that, given a CMSO-formula, it provides the homomorphism classes $\mathcal{C}$, the subset of accepting classes and the algorithms computing the classes of base graphs as well as the update functions for the regular property $\mathcal{P}$ on $(t+1)$-terminal recursive graphs. The regularity is actually proven in [11] for all properties expressible by CMSO-formulae, which allows an arbitrary number of free variables over vertices, edges, vertex sets and and edge sets. For our applications, it is sufficient to consider properties over graphs and one vertex set, corresponding to formulae with a unique free variable, which is a set of vertices.

To our knowledge, the question whether all regular properties are CMSO-expressible is still open.

2.2 Treewidth, minimal triangulations and potential maximal cliques

Chordal graphs and clique trees A graph H is *chordal* (or *triangulated*) if every cycle of length at least four has a chord, i.e., an edge between two nonconsecutive vertices of the cycle. By a classical result due to Buneman and Gavril [16, 40], every chordal graph G has a tree decomposition such that each bag of the decomposition is a maximal clique of G. Such a tree decomposition is referred as a *clique tree* of the chordal graph G.

Minimal triangulations, potential maximal cliques and minimal separators A *triangulation* of a graph $G = (V, E)$ is a chordal graph $H = (V, E')$ such that $E \subseteq E'$. Graph H is a *minimal triangulation* of G if for every edge set E'' with $E \subseteq E'' \subset E'$, the graph $F = (V, E'')$ is not chordal. It is well known that for any graph G, $\mathrm{tw}(G) \leq k$ if and only if there is a triangulation H of G of clique size at most $k+1$.

Let u and v be two non adjacent vertices of a graph G. A set of vertices $S \subseteq V$ is a *u,v-separator* if u and v are in different connected components of the graph $G[V(G) \setminus S]$. A connected component C of $G[V(G) \setminus S]$ is a *full component associated to* S if $N(C) = S$. Separator S is a *minimal u,v-separator* of G if no proper subset of S is a u,v-separator. Notice that a minimal separator can be strictly included in another one, if they are minimal separators for different pairs of vertices. If G is chordal, then for any minimal separator S and any clique tree T_G of G there is an edge e of T_G such that S is the intersection of the maximal cliques corresponding to endpoints of e [16, 40]. We say that S *corresponds* to e in T_G.

We will need the following result of Berry et al. [5].

Proposition 3 ([5]). *There is an algorithm listing the set Δ_G of all minimal separators of an input graph G in time $\mathcal{O}(n^3|\Delta_G|)$.*

A set of vertices $\Omega \subseteq V(G)$ of a graph G is called a *potential maximal clique* if there is a minimal triangulation H of G such that Ω is a maximal clique of H.

Proposition 4 ([13]). *Let Π_G denote the set of all potential maximal cliques of graph G. We have $|\Pi_G| \leq n|\Delta_G|^2 + n|\Delta_G| + 1$, and the set Π_G can be listed in time $\mathcal{O}(n^2 m|\Delta_G|^2)$.*

We also have:

Proposition 5 ([35]). *The set of potential maximal cliques can be listed in time $O(1.7347^n)$.*

Let Ω be a potential maximal clique. By [12], a subset $S \subseteq \Omega$ is a minimal separator of G if and only if S is the neighborhood of a connected component of $G[V(G) \setminus \Omega]$.

For a minimal separator S and a full connected component C of $G[V(G) \setminus S]$, we say that (S, C) is a *full block* associated to S. We sometimes use the notation (S, C) to denote the set of vertices $S \cup C$ of the block. It is easy to see that if $X \subseteq V$ corresponds to the set of vertices of a block, then this block (S, C) is unique: indeed, $S = N(V \setminus X)$ and $C = X \setminus S$. For convenience, the couple $(\emptyset, V)$ is also considered as a full block. For a minimal separator S, a full block (S, C), and a potential maximal clique Ω, we call the triple (S, C, Ω) *good* if $S \subseteq \Omega \subseteq C \cup S$. By [32], the number of good triples is at most $n|\Pi_G|$.

The following proposition was obtained by Fomin and Villanger [35].

Proposition 6 ([35]). *Let $G[F]$ be an induced subgraph of a graph G, let TF be a minimal triangulation of $G[F]$. There exists a minimal triangulation TG of G such that TF is an induced subgraph of TG.*

Equivalently, for every clique K_G of TG, the set $K_G \cap F$ is a (possibly empty) clique of TF.

Moreover, they consider the problem of finding a maximum induced subgraph of treewidth at most t:

Proposition 7 ([35]). *Given a graph G and with its set Π_G of potential maximal cliques, problem* Maximum Induced Subgraph of Treewidth $\leq t$ *can be solved in time $O(|\Pi_G| n^{t+4})$.*

By Propositions 7, 4 and 5, we deduce that for fixed t the problem can be solved in $O(1.7347^n)$ time for arbitrary graphs, and in polynomial time for classes of graphs with polynomial number of minimal separators.

3 Optimal induced subgraph for $\mathcal{P}$ and t

Let $t \geq 0$ be an integer and $\mathcal{P}(G, X)$ be a property. We define the following generic problem.

Optimal Induced Subgraph for $\mathcal{P}$ and t **Input:** A graph G **Task:** Find sets $X \subseteq F \subseteq V$ such that X is of maximum size, the induced subgraph $G[F]$ is of treewidth at most t and $\mathcal{P}(G[F], X)$ is true.

Let us give two examples of problems that are particular cases of Optimal Induced Subgraph for $\mathcal{P}$ and t, when $\mathcal{P}(G, X)$ is a regular property.

1. Let $\mathcal{F}$ be a finite family of graphs containing at least one planar graph. The problem Maximum induced $\mathcal{F}$-minor free graph takes as input a graph G and asks for an induced subgraph $G[F]$ such that $G[F]$ contains no minor from $\mathcal{F}$, and F is of maximum size for this property. As we shall see in details in Section 4, the property $\mathcal{P}(G[F], X)$ expressing the fact that $G[F]$ is $\mathcal{F}$-minor free and $X = F$ is the vertex set of $G[F]$ can be expressed by a CMSO. Since $\mathcal{F}$ contains a planar graph, $G[F]$ must be of treewidth at most t for some constant t depending only on $\mathcal{F}$ [64]. Therefore, this problem (or the equivalent problem Minimum $\mathcal{F}$-Deletion) is a particular case of Optimal Induced Subgraph for $\mathcal{P}$ and t.

2. The problem Independent $\mathcal{H}$-Packing was introduced by Cameron and Hell [18]. Here $\mathcal{H}$ denotes a finite set of connected graphs, and the task is to find, in an input graph G, a maximum number of disjoint copies of graphs from $\mathcal{H}$ such that there is no edges between the copies. Clearly these copies induce a subgraph $G[F]$ of bounded treewidth. We will give a CMSO-formula expressing the property $\mathcal{P}(G[F], X)$, which is true if and only if $G[F]$ is a collection of copies of $\mathcal{H}$, and X has exactly one vertex in each connected component of $G[F]$. This problem, generalizing the Maximum Induced Matching, is again a particular case of Optimal Induced Subgraph for $\mathcal{P}$ and t.

We prove here the main theorem of this article.

Theorem 1. *For any fixed t and any regular property $\mathcal{P}$, the problem* Optimal Induced Subgraph for $\mathcal{P}$ and t *is solvable in $|\Pi_G| n^{t+\mathcal{O}(1)}$ time, when Π_G is given in the input.*

Let us note that by Proposition 2, results of Theorem 1 hold for every property $\mathcal{P}(G, X)$ expressible by a CMSO-formula. Combined with Propositions 4 and 5, we obtain the following application of Theorem 1.

Corollary 1. *For any fixed t and regular property $\mathcal{P}$, problem* Optimal Induced Subgraph for $\mathcal{P}$ and t *can be solved in $\mathcal{O}(1.7347^n)$ time for arbitrary graphs, and in polynomial time for classes of graphs with polynomial number of minimal separators.*

3.1 Notations and data structures

Our algorithm proceeds by dynamic programming on blocks and on good triples. The general strategy of dynamic programming over blocks and good triples follows the ideas from [32] and [35] for computing the treewidth and subgraphs of bounded treewidth. However, the devil is in details, and we need more work to make this strategy applicable for our problem.

Recall that in our definition of $(t+1)$-terminal graphs, the set of terminals is ordered. The vertices of our graph are numbered from 1 to n. An ordered set W of vertices corresponds to this natural ordering over set W. Property $\mathcal{P}$ is regular, so notations $\mathcal{C}$, h and $\odot_f$ correspond to Definition 1.

Let $G[F]$ be an induced subgraph of G and let TF be a triangulation of $G[F]$. We say that a minimal triangulation TG of G *respects* TF if, for any clique K of TG, its intersection with F is a clique in TF. By Proposition 6, if $G[F]$ is of treewidth at most t, then there exists a (minimal) triangulation TF of $G[F]$ of width at most t, and a minimal triangulation TG of G respecting TF.

The next definition and the following notations are crucial for our algorithm.

Definition 2 (Partial Compatible Solution)**.** *Let (S, C) denote a full block and (S, C, Ω) denote a good triple. Let $W \subseteq S$ (resp. $W \subseteq \Omega$) be a vertex subset of size at most $t+1$ and $c \in \mathcal{C}$ be a homomorphism class for property $\mathcal{P}$. We say that $(G[F], X)$ is a* partial solution compatible with *(S, C, W, c) (resp. with (S, C, Ω, W, c)) if:*

1. *$F \subseteq S \cup C$ and $F \cap S = W$ (resp. $F \cap \Omega = W$);*
2. *the $(t+1)$-terminal recursive graph $H = (F, W, E(G[F]))$ satisfies $h(H, X) = c$;*
3. *there is a triangulation TF of $G[F]$ of width at most t and a minimal triangulation TG of G respecting TF, such that S is a minimal separator (resp. Ω is a maximal clique) of TG.*

The third condition implies that W is a clique in the triangulation TF of $G[F]$.

Let $\alpha(S, C, W, c)$ (resp. $\beta(S, C, \Omega, W, c)$) denote the size of a largest vertex subset X such that $(G[F], X)$ is a partial solution compatible with (S, C, W, c) (resp. compatible with (S, C, Ω, W, c)). Observe that in the β function, W represents the intersection between the partial solution and the potential maximal clique Ω, while in the definition of the α function, W is the intersection of the partial solution with the minimal separator S. If partial compatible solutions do not exist, we simply set α or β to $-\infty$.

3.2 The algorithm

Our algorithm proceeds by dynamic programming on full blocks and good triples. By [32], the number of good triples is $O(n|\Pi_G|)$. The blocks are first sorted by size. For each block (S, C) by increasing size, we first compute the values $\beta(S, C, \Omega, W, c)$ from values $\alpha(S_i, C_i, W_i, c_i)$ corresponding to smaller blocks, then we compute the values $\alpha(S, C, W, c)$

from values $\beta(S, C, \Omega, W', c')$, as described in Algorithm 1.

Algorithm 1: OPTIMAL INDUCED SUBGRAPH FOR $\mathcal{P}$ AND t

Input: graph G and Π_G

Output: sets $X \subseteq F \subseteq V(G)$ such that $G[F]$ has treewidth at most t, $\mathcal{P}(G[F], X)$ is true and X is of maximum size

Order all full blocks by inclusion;
forall the *full blocks* (S, C) *in this order* **do**
  forall the *good triples* (S, C, Ω), **all** $W \subseteq \Omega$ *of size* $\leq t+1$ *and* **all** $c \in \mathcal{C}$ **do**
    if $\Omega = S \cup C$ **then**
      Compute $\beta(S, C, \Omega, W, c)$ using Equation 2;
    else
      Compute $\beta(S, C, \Omega, W, c)$ using Equations 4, 5, 6, and 7 ;
  forall the $W \subseteq S$ *of size* $\leq t+1$ *and* **all** $c \in \mathcal{C}$ **do**
    Compute $\alpha(S, C, W, c)$ using Equation 3;
Compute the optimal solution using Equation 8;

Consider a $(t+1)$-terminal recursive graph $D = (V_D, T, E_D)$ and let c be a homomorphism class. Although this is not explicitly required by the definition of regular properties (Definition 1), we may assume w.l.o.g. that all sets Y such that $h(D, Y) = c$ have the same intersection with the set T of terminals. (Otherwise, if sets Y and Y' have different intersections with T but $h(D, Y) = h(D, Y') = c$, we can "split" class c in at most 2^{t+1} classes, one for each possible intersection between T and such a vertex subset Y.) Moreover the class c encodes the intersection of Y with the set of terminals of D, i.e., given the homomorphism class c, we can retrieve the rank of the vertices of $Y \cap T$.

Therefore we assume that we have a function $term(c, T)$, taking a class c and an ordered set T of terminals, and returning the terminals that belong to Y, for any Y such that $h(D, Y) = c$.

The base case. The base case consists in minimal full blocks (S, C, Ω), in which case $\Omega = S \cup C$ by [12]. In this situation, for any partial solution $(G[F], X)$ compatible with (S, C, Ω, W, c) we must have $F = W$, hence $G[W]$ corresponds to a base $(t+1)$-terminal graph. Also, we must have $X = term(c, W)$, so X is unique (or might not exist).

$$\beta(S, C, \Omega, W, c) = \begin{cases} |X| & \text{if there is } X \subseteq W \text{ such that } h(G[W], X) = c \\ -\infty & \text{otherwise} \end{cases} \tag{2}$$

The computation of each value $\beta(S, C, \Omega, W, c)$ corresponding to a base case takes $O(n)$ time, because we have to store the value in a table indexed by (S, C, Ω, c). The number of good triples is $O(n|\Pi_G|)$ so altogether these computations take $O(n^{t+3}|\Pi_G|)$ time. (Actually, one can prove by a more careful analysis that the number of good triples corresponding to base cases is at most n.)

Computing α from β. Our goal is to compute $\alpha(S, C, W, c)$ from values $\beta(S, C, \Omega, W', c')$ such that (S, Ω, C) is a good triple and $W = W' \cap S$.

Consider any partial solution $(G[F], X)$ compatible with (S, C, W, c). Let TF be a triangulation of $G[F]$ like in Definition 2 and let TG be a minimal triangulation of G respecting TF. Let Ω be the maximal clique of TG such that $S \subseteq \Omega \subseteq S \cup C$ (this clique is unique by [12]) and take $W' = \Omega \cap F$. Note that $(G[F], X)$ is also a partial compatible solution for (S, C, Ω, W', c') where c' is the homomorphism class of $h(H', X)$; here H' is the $(t+1)$-terminal recursive graph $(F, W', E(G[F]))$. Also observe that the $(t+1)$-terminal graph $H = (F, W, G[F])$ is obtained from H' by the unary composition operation $f(W', W)$ that consists in removing $W' \backslash W$ from the set of terminals, and possibly renumbering the remaining terminals. Therefore $\odot_{f(W',W)}(c') = c$.

We claim that:

Lemma 1.

$$\alpha(S, C, W, c) = \max \beta(S, C, \Omega, W', c'), \tag{3}$$

where the maximum is taken over potential maximal cliques Ω such that (S, C, Ω) is a good triple, all subsets $W' \subseteq \Omega$ of size at most $t+1$ such that $W' \cap S = W$ and all classes $c' \in \mathcal{C}$ such that $\odot_{f(W',W)}(c') = c$.

Proof. By the above observation, $\alpha(S, C, W, c)$ is at most the right-hand side of the equality. Conversely, let (S, C, Ω, W', c') be the quintuple realizing the maximum value of the right-hand side expression. Let $(G[F], X)$ be a partial solution compatible with (S, C, Ω, W', c'). Observe that $(G[F], X)$ is also a partial solution compatible with (S, C, W, c), hence $\alpha(S, C, W, c) \geq |X|$. This proves the correctness of the formula computing $\alpha(S, C, W, c)$. □

For computing all values $\alpha(S, C, W, c)$ from values $\beta(S, C, \Omega, W', c')$, we proceed in a slightly different and more efficient way than the one described in the Algorithm 1. When $\beta(S, C, \Omega, W', c')$ is computed (lines 5 or 7 of the algorithm), if $\odot_{f(W',W)}(c') = c$ we simply update the value of $\alpha(S, C, W, c)$ by taking the maximum between the previous value and $\beta(S, C, \Omega, W', c')$. This only costs an extra $O(n)$ for each quintuple (S, C, Ω, W', c'). The number of such quintuples is $\mathcal{O}(n^{t+2}|\Pi_G|)$, thus the total cost of these computations is $\mathcal{O}(n^{t+3}|\Pi_G|)$.

Computing β from α. We now compute $\beta(S, C, \Omega, W, c)$ from values $\alpha(S_i, C_i, W_i, c_i)$ where C_i, $1 \leq i \leq p$ are the connected components of $G[C \setminus \Omega]$, $S_i = N_G(C_i)$, $W_i = C_i \cap S_i$ and c_i are classes (still to be guessed). Recall that, by [12], (S_i, C_i) are full blocks.

Intuitively, let $(G[F], X)$ be an optimal partial solution for $\beta(S, C, \Omega, W, c)$. We denote by $H = (F, W, E_H)$ the $(t+1)$-terminal recursive graph corresponding to $G[F]$ with terminal set W, and let $H_i = (F_i, W_i, E_i)$ be its trace on the smaller block (S_i, C_i). Hence $F_i = F \cap (S_i \cup C_i)$, $W_i = W \cap S_i$ and $E_i = E(G[F_i])$. Also denote $X_i = X \cap (S_i \cup C_i)$. Observe that H is obtained from the smaller H_is as follows:

- on each H_i, we introduce the terminals of $W \setminus W_i$, obtaining a graph $H_i^+ = (F_i \cup W, W, E_i^+)$ with W as set of terminals and with $E_i^+ = E(G[F_i \cup W])$ as edge set.
- we perform a sequence of joins, gluing one by one $H_1^+, H_2^+, \ldots, H_p^+$ on the same set of terminals W.

Formally, let us first define $\delta_i(S, C, \Omega, W, c_i^+)$ to be the size of the largest partial solution $(G[F_i^+], X_i^+)$ compatible with (S, C, Ω, W, c_i^+) such that $F_i^+ \subseteq \Omega \cup C_i$. (This partial solution was denoted above by H_i^+, F_i^+ corresponds to $F_i \cup W$, and X_i^+ is $X_i \cup (X \cap W)$.) Consider the composition operation $in(W_i, W)$ which takes two $(t+1)$-terminal graphs, with terminal sets W_i and W respectively, and composes them into a new $(t+1)$-terminal graph having W as set of terminals. In the gluing operation, terminal number j of W_i is glued on terminal number k of W if and only if they correspond to the same vertex of G. Hence, this composition operation $in(W_i, W)$ only depends on W_i and W. Let $X_W \subseteq W$, let $G[W]$ denote the base $(t+1)$- having W as set of terminals, and c_W be the homomorphism class $h(G[W], X_W)$.

Lemma 2.

$$\delta_i(S, C, \Omega, W, c_i^+) = \max_{c_i, c_W \ s.t.\ \odot_{in(W_i,W)}(c_i, c_W) = c_i^+} \alpha(S_i, C_i, W_i, c_i) + |term(c_W, W) \setminus term(c_i, W_i)| \tag{4}$$

over all classes c_i and c_W such that $\odot_{in(W_i,W)}(c_i, c_W) = c_i^+$ and $c_W = h(G[W], X_W)$ for some $X_W \subseteq W$.

Proof. Let $(G[F_i^+], X_i^+)$ be a maximal partial solution compatible with (S, C, Ω, W, c_i^+) such that $F_i^+ \subseteq \Omega \cup C_i$. Denote $F_i = F_i^+ \cap (S_i \cup C_i)$, $X_i = X_i^+ \cap (S_i \cup C_i)$, $X_W = X \cap W$. Observe that $(G[F_i], X_i)$ is a partial solution compatible with (S_i, C_i, W_i, c_i) for some class c_i, that $c_W = h(G[W], X_W)$, and these classes must satisfy $\odot_{in(W_i,W)}(c_i, c_W) = c_i^+$. Hence $\delta_i(S, \Omega, C, W, c_i^+)$ is at most equal to the right-hand side of the equation (note that $term(c_W, W) \setminus term(c_i, W_i) = X_i^+ \setminus X_i$).

Conversely, let c_i, c_W be the classes maximizing the right-hand side of the equation. Take a maximum partial solution $(G[F_i], X_i)$ contained in $S_i \cup C_i$, compatible with (S_i, C_i, W_i, c_i), where $\odot_{in(W_i,W)}(c_i, c_W) = c_i^+$. Then the graph $(F_i \cup W, W, E(G[F_i \cup W]))$ together with the vertex subset $X_i \cup term(c_W, W)$ is a partial solution compatible with (S, C, Ω, W, c_i^+), and the equality follows. □

We introduce another notation $\gamma_i(S, C, \Omega, W, c)$, corresponding to the largest partial solution compatible with (S, C, Ω, W, c), contained into $\Omega \cup C_1 \cup \cdots \cup C_i$. It corresponds to the gluing of some partial solutions $(H_1^+, X_1^+), \ldots (H_i^+, X_i^+)$.

Lemma 3. *Function γ_i is computed as follows.*

$$\gamma_1(S, C, \Omega, W, c) = \delta_1(S, \Omega, C, W, c) \tag{5}$$

For any i, $2 \leq i \leq p$,

$$\gamma_i(S, C, \Omega, W, c) = \max_{c', c''} \gamma_{i-1}(S, C, \Omega, W, c') + \delta_i(S, \Omega, C, W, c'') - |term(c', W)|, \tag{6}$$

over all characteristics $c', c'' \in \mathcal{C}$ such that $\odot_{g(W)}(c', c'') = c$, where $g(W)$ is the composition operation corresponding to a join operation on W. I.e., the matrix $m(g(W))$ of $g(W)$ has $|W|$ rows, and $m_{j,1}(g(W)) = m_{j,2}(g(W)) = j$ for each row j.

Proof. The proof is trivial for γ_1.

Now for any $F \subseteq \Omega \cup C_1 \cup \cdots \cup C_i$, note that $(G[F], X)$ is a partial solution compatible with (S, C, Ω, W, c) if and only if $(G[F \setminus C_i], X \setminus C_i)$ (resp. $(G[F \setminus (C_1 \cup \cdots \cup C_{i-1})], X \setminus (C_1 \cup \cdots \cup C_{i-1})))$ are partial solutions compatible with (S, Ω, C, W, c'') (resp. (S, C, Ω, W, c')) and $\odot_{g(W)}(c', c'') = c$. The term $|term(c', W)|$ corresponds to $X \cap W$ and avoids over-counting of these vertices. □

The following result is a direct consequence of the definition of β and γ functions.

Lemma 4.

$$\beta(S, C, \Omega, W, c) = \gamma_p(S, C, \Omega, W, c). \tag{7}$$

We claim that computing, for a fixed quadruple (S, C, Ω, W), the values $\beta(S, C, \Omega, W, c)$ from values α, takes $O(n^2)$ time. Again by [32], the smaller blocks (S_i, C_i) can be listed in $O(m)$ time. For each i, the computation of function $\delta_i(S, \Omega, C, W, c_i^+)$ takes $O(|S_i| + |C_i|)$ time, because we need to access the values $\alpha(S_i, W_i, C_i, c_i)$. The sum of these values is at most $n + m$ [32]. Computing $\gamma_i(S, C, \Omega, W, c)$ from values γ_{i-1} and δ_i can be done in $O(n)$ time for each i.

Therefore the running time of the algorithm is the number of quintuples (S, C, Ω, W, c) times n^2, that is $O(|\Pi_G| n^{t+4})$.

The global solution. It can be obtained by considering the (special) full block $(\emptyset, V)$.

Lemma 5. *The solution size is*

$$\max_c \alpha(\emptyset, V, \emptyset, c), \tag{8}$$

over all accepting classes c, i.e., classes such that $(h(G, X) = c)$ implies that $\mathcal{P}(G, X)$.

Proof. By definition of regular properties and of $\alpha(\emptyset, V, \emptyset, c)$, our problem has a solution of size at least $\max_c \alpha(\emptyset, V, \emptyset, c)$ over accepting classes c.

Let $(G[F], X)$ be a maximum size solution for our problem. By Proposition 6, this solution is compatible with $\alpha(\emptyset, V, \emptyset, c)$ for the class c of the $(t + 1)$-terminal graph graph $(F, \emptyset, E(G[F]))$, which achieves the proof of the lemma. $\square$

This latter computation takes constant time.

The total running time of the algorithm is $O(|\Pi_G| n^{t+4})$. Note that, instead of keeping the size of the largest solution $(G[F], X)$, we could explicitly store the vertex subsets (F, X) of G.

3.3 Extensions

Theorem 1 can be extended to *weighted* and *annotated* versions of problem Optimal Induced Subgraph for $\mathcal{P}$ and t, for any $t \geq 0$ and any regular property $\mathcal{P}$.

> Optimal Weighted Annotated Induced Subgraph for $\mathcal{P}$ and t
> **Input:** A graph $G = (V, E)$ a weight function $w : V \to \mathbb{R}$, a set $U \subseteq V$ of annotated vertices and a number t.
> **Task:** Find sets $X \subseteq F \subseteq V$ such that F contains U, the induced subgraph $G[F]$ is of treewidth at most t, property $\mathcal{P}(G[F], X)$ is true and X is of maximum weight under these conditions.

Theorem 2. *For any fixed t and any regular property $\mathcal{P}$, the problem* Optimal Weighted Annotated Induced Subgraph for $\mathcal{P}$ and t *is solvable in $|\Pi_G| n^{O(1)}$ time, when Π_G is given in the input.*

In particular the problem can be solved in $\mathcal{O}(1.7347^n)$ time for arbitrary graphs, and in polynomial time for classes of graphs with polynomial number of minimal separators.

For this purpose, we slightly adapt the definitions of α and β functions. In order to force the annotated vertices to be in F, each value $\alpha(S, C, W, c)$ (resp. $\beta(S, C, \Omega, W, c)$ such that $U \cap S \nsubseteq W$ (resp. $U \cap \Omega \nsubseteq W$) is immediately set to $-\infty$, meaning that such a partial solution is rejected.

In order to maximize the weight of the solution, the values $\alpha(S, C, W, c)$ (respectively $\beta(S, C, \Omega, W, c)$) will correspond to the maximum weight over partial solutions compatible with (S, C, W, c) (resp. (S, C, Ω, W, c)). In the algorithm, we simply replace the cardinality of sets (e.g., $|X|$ in Equation 2, $|term(c', W)|$ is Equation 6 and $|term(c_W, W) \setminus term(c_i, W_i)|$ in Equation 4) by the weights of these sets.

We also point out that the weights can be negative. In particular, we can use Theorem 2 to compute an induced subgraph $G[F]$ of treewidth at most t and a subset $X \subseteq F$ such that $\mathcal{P}(G[F], X)$ is true, and X is of minimum size (or weight) under these conditions.

One can imagine more extensions of Theorems 1 and 2. A natural one consists in finding sets X and F such that the size of X is *exactly* an input value v. For this purpose, we can adapt our definitions of α and β to store, for each possible value $v' \leq v$, a boolean $\alpha(S, C, W, c, v')$ (resp. $\beta(S, C, \Omega, W, c, v')$), set to *true* if and only if there exists partial solution $(G[F'], X')$ compatible with (S, C, W, c) (resp. (S, C, Ω, W, c) such the size of X' is exactly v'. The computation of α and β is quite straightforward, by adapting Equations 2 to 8. The complexity of the algorithm is multiplied by a polynomial factor.

Even more involved, we can consider properties $\mathcal{P}(G, X_1, \ldots, X_p, E_1, \ldots, E_q)$, where each X_i is a vertex subset and each E_j is an edge subset of graph G. The notion of regularity extends in a very natural way to several variables. Recall that Borie *et al.* [11] proved that *all* properties expressible by CMSO-formulae are regular, so we are allowed to use any (fixed) number of free variables corresponding to vertex sets and edge sets.

Let $t \geq 0$ be an integer and $\mathcal{P}(G, X_1, \ldots, X_p, E_1, \ldots, E_q)$ be a regular property on graphs and vertex subsets X_i and edge subsets E_j. We define the following generic problem.

Constrained Induced Subgraph for $\mathcal{P}$ and t

Input: A graph G, integer values $v_1, \ldots, v_p \leq n$ and $w_1, \ldots, w_p \leq \frac{n(n-1)}{2}$
Task: Find $F \subseteq V$, sets $X_i \subseteq F$ and $E_j \subseteq E(G[F])$ such that the induced subgraph $G[F]$ is of treewidth at most t, $\mathcal{P}(G, X_1, \ldots, X_p, E_1, \ldots, E_q)$ is true, each set X_i is of size v_i and each set E_j is of size w_j.

Since property $\mathcal{P}$ is regular, we need to adapt the definition of partial solutions to more variables (again very naturally) and then we define as above boolean functions

$$\alpha(S, C, W, c, v'_1, \ldots, v'_p, w'_1 \ldots, w'_q),$$

respectively

$$\beta(S, C, \Omega, W, c, v'_1, \ldots, v'_p, w'_1 \ldots, w'_q)$$

to be *true* if there exists a partial solution $(G[F'], X'_1, \ldots, X'_p, E'_1, \ldots, E'_q)$ compatible with (S, C, W, c) (resp. (S, C, Ω, W, c)) such that each X'_i is of size v'_i and each E'_j is of size w'_j. For computing the α and β values, we must again adapt Equations 2 to 8. Basically, for each class c, the function $term(c, W)$ used in the equations for a homomorphism class c and an order set of terminals W must now return each intersection of type $X'_i \cap W$ for vertex sets and $E'_j \cap G[W]$ for edge sets. These intersections will be used to avoid overcounting when glueing partial solutions. The complexity of the algorithm becomes larger by a factor of $n^{\mathcal{O}(p+q)}$.

Therefore we can solve problems like finding, among maximum induced subgraph of treewidth at most t, the one with minimum dominating set.

4 Applications

In this section we discuss several applications of Theorem 1. Our results are summarized in the following theorem. Recall that the problems have been defined in the *Introduction*.

Theorem 3. *Let G be an n-vertex graph given together with the set of its potential maximal cliques Π_G. Then*

- Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles*,*
- Maximum Induced Subgraph with $\leq \ell$ copies of p-cycles*,*
- Maximum Induced Subgraph with $\leq \ell$ copies of Minor Models from $\mathcal{F}$*, where $\mathcal{F}$ contains a planar graph,*
- Independent $\mathcal{G}(t,\varphi)$-Packing*, and*
- k-in-a-Graph From $\mathcal{G}(t,\varphi)$

are solvable in time $|\Pi_G| \cdot n^{\mathcal{O}(1)}$. Here the hidden constants in $\mathcal{O}$ depend on m, p, ℓ, $\mathcal{F}$, t, and φ.

Combined with Proposition 5, Theorem 3 implies the following.

Corollary 2. *Let G be an n-vertex graph. All problems from Theorem 3 are solvable in time $\mathcal{O}(1.7347^n)$.*

The proof of Theorem 3 follows from Theorem 1 and Lemmata 6, 7, 8, 9, and 10.

Let us remark that Theorem 3 also holds for different modifications of these problems, like requirements of the maximum induced subgraph being connected, of maximum vertex degree at most some constant Δ, etc. Such modifications easily capture problems like computing a longest induced path, cycle, or an induced tree with given maximum vertex degree.

Hitting and packing cycles of length $0 \pmod{m}$. We will need the following result of Thomassen.

Proposition 8 ([70])**.** *For every integers $\ell, m > 0$ there exists an integer $k(\ell, m) > 0$ such that the treewidth of a graph with at most ℓ vertex-disjoint cycles from $\mathcal{F}_m$ is at most $k(\ell, m)$.*

With the help of Proposition 8, we obtain the following lemma.

Lemma 6. Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles *is a special case of* Optimal Induced Subgraph for $\mathcal{P}$ and t *with $t = f(\ell, m)$, where f depends only on m and ℓ.*

Proof. For a graph G let F be the maximum vertex set such that $G[F]$ has at most ℓ vertex-disjoint cycles from $\mathcal{F}_m$. We put $f(\ell, m) = k(\ell, m)$, where $k(\ell, m)$ is the integer from Proposition 8. By Proposition 8, the treewidth of G_F is at most $f(\ell, m)$.

Then Maximum Induced Subgraph with $\leq \ell$ copies of $\mathcal{F}_m$-cycles is to maximize $|X|$ for the following property

$$\mathcal{P}(G[F], X) = \{F = X \text{ and } G[F] \text{ contains at most } \ell \text{ vertex-disjoint cycles from } \mathcal{F}_m.\}$$

To show that $\mathcal{P}(G[F], X)$ is regular, we observe that it is expressible by a CMSO-formula. Indeed, this formula expresses that for every partition of $V(G_F)$ into $\ell + 1$ subsets, there is a subset containing no cycle from $\mathcal{F}_m$. □

Hitting long cycles. We need the following result, which is due to Birmelé et al.

Proposition 9 ([7])**.** *Graphs without ℓ disjoint cycles of length at least p are of treewidth $\mathcal{O}(\ell^2 p)$.*

By making use of Proposition 9, it is easy to prove the following lemma.

Lemma 7. Maximum Induced Subgraph with $\leq \ell$ copies of p-cycles *is a special case of* Optimal Induced Subgraph for $\mathcal{P}$ and t *with* $t = \mathcal{O}(\ell^2 p)$.

Proof. For a graph G let F be the maximum vertex set such that $G[F]$ has at most ℓ vertex-disjoint cycles of length at least p. By Proposition 9, the treewidth of $G[F]$ is at most $\mathcal{O}(\ell^2 p)$. Then we are maximizing $|X|$ for the following property

$$\mathcal{P}(G[F], X) = \{F = X \text{ and } G[F] \text{ contains} \leq \ell \text{ vertex-disjoint cycles of length } \geq p.\}$$

To show that this property is regular, we observe that property of not having a cycle of length at least p is expressible in CMSO. Indeed, a property of a set C of vertices to induce a cycle is CMSO, and because p is fixed, the formula expressing the sentence that for every subset C inducing a cycle, the number of elements is at most p, is of constant length. Because ℓ is also fixed, it is possible to express by a constant size CMSO-formula the sentence that for every partition in $\ell + 1$ subsets there is a subset inducing a subgraph without a cycle of length at least p. □

Excluding planar minors. The following proposition follows almost directly from the excluded grid theorem of Robertson and Seymour [64], see also [65].

Proposition 10 ([64])**.** *For every integer $\ell > 0$ and family $\mathcal{F}$ containing a planar graph, there exists an integer $k(\ell, \mathcal{F}) > 0$ such that the treewidth of a graph with at most ℓ vertex-disjoint minor models from $\mathcal{F}$ is at most $k(\ell, \mathcal{F})$.*

Lemma 8. *If $\mathcal{F}$ contains a planar graph, then* Maximum Induced Subgraph with $\leq \ell$ copies of Minor Models from $\mathcal{F}$ *is a special case of* Optimal Induced Subgraph for $\mathcal{P}$ and t *with* $t = k(\ell, \mathcal{F})$.

Proof. For a graph G let F be the maximum vertex set such that $G[F]$ has at most ℓ vertex-disjoint models of minors from $\mathcal{F}$. By Proposition 10, the treewidth of $G[F]$ is at most $k(\ell, \mathcal{F})$. The property that a graph does not contain a fixed graph as a minor is known to be expressible in CMSO. This implies that the property

$$\mathcal{P}(G[F], X) = \{F = X \text{ and } G[F] \text{ has} \leq \ell \text{ vertex-disjoint minor models from } \mathcal{F}\}$$

is regular. □

Independent packing.

Lemma 9. INDEPENDENT $\mathcal{G}(t,\varphi)$-PACKING *is a special case of* OPTIMAL INDUCED SUBGRAPH FOR $\mathcal{P}$ AND t.

Proof. For a graph G let F be a vertex set such that $G_F = G[F]$ has the maximum number of connected components, and each of the components is in $\mathcal{G}(t,\varphi)$. Because the treewidth of every component does not exceed t, the treewidth of $G[F]$ does not exceed t. We use $cc(G[F])$ to denote the set of connected components of $G[F]$. Because the property that every connected component has regular is also regular, we have that the following property is regular

$$\mathcal{P}(G[F],X) = \{[X \subseteq V(G_F)] \wedge [\forall C \in cc(G_F)(C \in \mathcal{G}(t,\varphi) \wedge |X \cap C| = 1)].\}$$

□

***k*-in-a-graph.** Because in k-IN-A-GRAPH FROM $\mathcal{G}(t,\varphi)$, k is part of the input we need the annotated variant of the main theorem (Theorem 2). The following lemma follows from the definition of the problems.

Lemma 10. k-IN-A-GRAPH FROM $\mathcal{G}(t,\varphi)$ *is a special case of* OPTIMAL WEIGHTED ANNOTATED INDUCED SUBGRAPH FOR $\mathcal{P}$ AND t.

5 Graph classes

In this section we discuss the consequences of Theorem 3 for special graph classes. In particular, by Proposition 4, every class of graphs with polynomially many minimal separators also has polynomially many potential maximal cliques. For example, every n-vertex *weakly chordal* graph, i.e. graph with no induced cycle or its complement of length greater than four, has $\mathcal{O}(n^2)$ minimal separators [12]. This class of graphs is a generalization of many graph classes intensively studied in the literature like chordal, split, and interval graphs. Another class of graphs of this type is the class of *circular-arc* graphs, intersection graphs of a set of arcs on the circle. Every circular-arc with n vertices has at most $2n^2 - 3n$ minimal separators [52]. The class of d-trapezoid graphs is defined as follows. Let $L_1, \dots, L_d$ be d parallel lines in the plane. A d-trapezoid is the polygon obtained by choosing an interval I_i on every line L_i and connecting the left, respectively, right endpoint of I_i with the left, respectively, right endpoint of I_{i+1}. A graph is a *d-trapezoid graph* if it has an intersection model consisting of d-trapezoids between d parallel lines. Every d-trapezoid graph has at most $(2n-3)^{d-1}$ minimal separators [55], see also [15]. An intersection graph of polygons enclosed by a bounding circle is is know as a *polygon-circle graph.* As it was observed by Suchan in [68], every polygon-circle with n vertices has $\mathcal{O}(n^2)$ minimal separators. See Fig 1 of the *Introduction* for the relations between most known classes of graphs with polynomially many minimal separators. We refer to the encyclopedia of graph classes [15] for definitions of different graphs from Fig 1.

Let us remark that the only information for our algorithms we need is the bound on the number of minimal separators in the specific graph class. While many of the algorithms from the literature for intersection classes of graphs strongly use the intersection model this is not necessary for our algorithms—they produce correct output regardless of whether the input actually belongs to the specific class of graphs. If the number of minimal separators and thus potential maximal cliques is bounded, our algorithm correctly solves the problem. Otherwise, the algorithm correctly reports that the given input is not from the restricted domain. Such type of algorithms were called *robust* by Raghavan and Spinrad [62]. For example, while recognition of d-trapezoid and polygon-circle graphs is NP-complete [72, 61], our algorithm either correctly solves the problem or outputs that the input graph is not d-trapezoid or polygon-circle.

Corollary 3. *All problems from Theorem 3 are solvable in polynomial time on classes of graphs from Fig 1.*

On several classes of graphs even more general problems can be solved. The observation here is that for many classes of graphs from Fig 1, the treewidth of a graph is upper bounded by some function of other parameters like the maximum clique-size or maximum degree.

For example, Yannakakis and Gavril [73] have shown that for every fixed χ, a maximum induced subgraph of a chordal graph colorable in χ colors can be found in polynomial time. To see why this result follows as a corollary of our theorem, let us observe that for chordal graphs, as for all perfect graphs, the chromatic number is equal to the maximum clique size, see e.g. [44]. On the other hand, the treewidth of a chordal graph is known to be equal to the maximum clique size minus one. Thus every induced χ-colorable subgraph of a chordal graph is of treewidth at most $\chi - 1$. Since colorability in a constant number of colors is expressible in CMSO, the result follows.

For other variant of colorings, we need the the following proposition due to Gaspers et al.

Proposition 11 ([39])**.** *Let G be a graph of maximum vertex degree at most D. Then the treewidth of G is at most*

- $4D$*, if G is a circle graph,*
- $2D$*, if G is a weakly chordal graph or a circular-arc graph.*

Combined with Proposition 11, Theorem 3 allows us to show that on several graph classes, in addition to problems encompassed by Corollary 3, even larger class of problems can be solved efficiently. For example, *edge coloring* of a graph is an assignment of colors to the edges of the graph so that no two adjacent edges have the same color. The *chromatic index* of a graph is the minimum number of colors required for edge coloring. By Vizing's theorem, for every graph with maximum vertex degree D, its chromatic index is either D or $D + 1$. Since edge coloring in a constant number of colors is expressible in CMSO, we conclude that the problem of finding a maximum induced edge-colorable in k colors subgraph (for a fixed constant k) is solvable in polynomial time on circle, weakly chordal and circular-arc graphs. Similarly, the problems like for a fixed constant k finding a maximum induced (connected) subgraph of maximum vertex degree at most k are also solvable in polynomial time on these classes of graphs.

The next lemma provides a different set of applications of the main theorem for special graph classes.

Lemma 11. *Let G be a graph excluding some fixed graph H as a minor. Then the treewidth of G is at most*

- $f(H)$ *for some function f of H only, if G is a weakly-chordal graph, and*

- $3|V(H)|$ *if G is a circular-arc graph.*

Proof. Let G be a weakly chordal graph excluding H as a minor. By a theorem from [29], there is a constant c_H such that every H-minor-free graph of treewidth at least $c_H k^2$ can be transformed by making only edge contractions either to a planar triangulation Γ_k of a $(k \times k)$-grid, or to Π_k, which is a graph obtained from G_k by adding a universal vertex. Since both Γ_k and Π_k for $k \geq 3$ contain an induced cycle of length at least 6, we conclude that the treewidth of G does not exceed some constant depending only on H. Indeed, otherwise a contraction of G, and hence G too, would contain an induced cycle of length more than 4.

For circular-arc graphs, we can prove the statement of the lemma by using the observation from [52] that every potential maximal clique of a circular-arc graph is the union of at most of three cliques. Thus every circular-arc graph of treewidth at least $3|V(H)|$ should contain a potential maximal clique of size at least $3|V(H)|$, and hence a clique of size at least $|V(H)|$. Thus every circular-arc graph of treewidth at least $3|V(H)|$ contains H as a minor. □

By combining Lemma 11 with Theorem 3, we obtain that Minimum $\mathcal{F}$-Deletion is solvable in polynomial time on circular-arc and weakly chordal graphs for every finite family $\mathcal{F}$ of graphs. The requirement that $\mathcal{F}$ contains a planar graph can be omitted in this case.

6 Conclusion

While regular properties and CMSO capture many interesting problems, it seems that the approach based on minimal triangulations is not restricted by these settings. Take for example the following problem.

> Minimum Induced Disjoint Connected ℓ-Subgraphs
> **Input:** A graph G, and a collection $\{T_1, T_2, \ldots, T_p\}$ of terminal vertices, $T_i \subseteq V(G)$, of size at most ℓ.
> **Task:** Find a set $F \subseteq V(G)$ of minimum size such that $G[F]$ has connected components $C_1, C_2, \ldots, C_p$ and for every $1 \leq i \leq p$, $T_i \subseteq C_i$.

This problem is a generalization of the Induced Disjoint Paths, where for a given set of p pairs of terminals x_i, y_i, $1 \leq i \leq p$, the task is to find a set of paths connecting terminals such that the vertices from different paths are not adjacent. Belmonte et al. [4] have shown that Induced Disjoint Paths is solvable in polynomial time on chordal graphs. Because p is part of the input and not fixed, this problem cannot be expressed by a CMSO-formula of constant size. On the other hand, by applying a modification of the dynamic programming algorithm over potential maximal cliques and minimal separators, it is possible to show that this problem is solvable in time proportional to the number of potential maximal cliques, up to polynomial factor $n^{t+\mathcal{O}(1)}$.

Another example can be the following problem. Let t be an integer.

> Homomorphism from t-Treewidth Subgraph
> **Input:** Graph G and H
> **Task:** Find a set $F \subseteq V(G)$ of maximum size such that the treewidth of $G[F]$ is at most t and there is a homomorphism from $G[F]$ to H.

By the classical result of Yannakakis and Gavril [73], for every fixed χ, a maximum induced subgraph of a chordal graph colorable in χ colors can be found in polynomial time.

Because coloring into χ colors is homomorphism in a complete graph on χ vertices, and because the treewidth of a χ-colorable chordal graph is at most $\chi - 1$, Homomorphism from t-Treewidth Subgraph extends this problem. However, the property of having a homomorphism to H is not CMSO-expressible because H is part of the input. Moreover, it is easy to see that already very special case of graph homomorphism problem, where we are asked for a homomorphism from a clique of size k (and thus of treewidth $k - 1$) to H is equivalent to deciding if H has a clique of size at least k, which is W[1]-hard. Thus homomorphism from G to H parameterized by the treewidth of G is W[1]-hard. But on the other hand, dynamic programming over potential maximal cliques and minimal separators shows that Homomorphism from t-Treewidth Subgraph is solvable in time proportional to the number of potential maximal cliques, up to polynomial factor $n^{O(t)}$.

Both examples indicate that even more general framework capturing problems solvable in time proportional to the number of potential maximal cliques can exist. Defining such a general framework is an interesting open question.

Another open question concerns counting problems. Our approach does not work for counting problems due to potential double counting in the process of computing functions α and β. We do not exclude a possibility that with additional (clever) ideas the main algorithm of the paper can also count maximum sets with regular properties but we do not know how to do it, and leave it as an interesting open question.

Another problem which seems to be very much related but still cannot be handled directly by our approach is Connected Feedback Vertex Set, where we are asked to find a minimum feedback vertex set inducing a connected subgraph. Interestingly, out approach works without problems for Maximum Induced Tree, where the task is to find a minimum feedback vertex set such the remaining graph is connected, i.e. a tree.

Acknowledgements We thank Bruno Courcelle, Daniel Lokshtanov, Mamadou Kanté, Dieter Kratsch, Saket Saurabh, Bich Dao and Dimitrios M. Thilikos for fruitful discussions and useful suggestions on the topic of the paper.

References

[1] S. R. Arikati and U. N. Peled, *A linear algorithm for the group path problem on chordal graphs*, Discrete Applied Mathematics, 44 (1993), pp. 185–190.

[2] S. R. Arikati, C. P. Rangan, and G. K. Manacher, *Efficient reduction for path problems on circular-arc graphs*, BIT, 31 (1991), pp. 182–193.

[3] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, Journal of Algorithms, 12 (1991), pp. 308–340.

[4] R. Belmonte, P. A. Golovach, P. Heggernes, P. van 't Hof, M. Kaminski, and D. Paulusma, *Finding contractions and induced minors in chordal graphs via disjoint paths*, in Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC 2011), vol. 7074 of Lecture Notes in Computer Science, Springer, 2011, pp. 110–119.

[5] A. Berry, J. P. Bordat, and O. Cogis, *Generating all the minimal separators of a graph*, Int. J. Found. Comput. Sci., 11 (2000), pp. 397–403.

[6] D. Bienstock, *On the complexity of testing for odd holes and induced odd paths*, Discrete Mathematics, 90 (1991), pp. 85–92.

[7] E. Birmelé, J. A. Bondy, and B. A. Reed, *The Erdős-Pósa property for long circuits*, Combinatorica, 27 (2007), pp. 135–145.

[8] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theor. Comput. Sci., 209 (1998), pp. 1–45.

[9] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos, *(Meta) kernelization*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, (FOCS 2009), Atlanta, Georgia, 2009, IEEE, pp. 629–638.

[10] H. L. Bodlaender, T. Kloks, and D. Kratsch, *Treewidth and pathwidth of permutation graphs*, SIAM J. Discrete Math., 8 (1995), pp. 606–616.

[11] R. B. Borie, R. G. Parker, and C. A. Tovey, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, Algorithmica, 7 (1992), pp. 555–581.

[12] V. Bouchitté and I. Todinca, *Treewidth and minimum fill-in: Grouping the minimal separators*, SIAM J. Comput., 31 (2001), pp. 212–232.

[13] ——, *Listing all potential maximal cliques of a graph*, Theor. Comput. Sci., 276 (2002), pp. 17–32.

[14] N. Bourgeois, B. Escoffier, V. T. Paschos, and J. M. M. van Rooij, *Fast algorithms for max independent set*, Algorithmica, 62 (2012), pp. 382–415.

[15] A. Brandstädt, V. Le, and J. P. Spinrad, *Graph Classes. A Survey*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, USA, 1999.

[16] P. Buneman, *A characterization of rigid circuit graphs*, Discr. Math., 9 (1974), pp. 205–212.

[17] K. Cameron, *Induced matchings*, Discrete Applied Mathematics, 24 (1989), pp. 97–102.

[18] K. Cameron and P. Hell, *Independent packings in structured graphs*, Math. Program., 105 (2006), pp. 201–213.

[19] K. Cameron, R. Sritharan, and Y. Tang, *Finding a maximum induced matching in weakly chordal graphs*, Discrete Mathematics, 266 (2003), pp. 133–142.

[20] J.-M. Chang, *Induced matchings in asteroidal triple-free graphs*, Discrete Applied Mathematics, 132 (2003), pp. 67–78.

[21] M. Chudnovsky and P. D. Seymour, *The three-in-a-tree problem*, Combinatorica, 30 (2010), pp. 387–417.

[22] B. Courcelle, *The monadic second-order logic of graphs I: Recognizable sets of finite graphs*, Inform. and Comput., 85 (1990), pp. 12–75.

[23] B. Courcelle, *The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues*, RAIRO Inform. Théor. Appl., 26 (1992), pp. 257–286.

[24] ——, *The expression of graph properties and graph transformations in monadic second-order logic*, in Handbook of graph grammars and computing by graph transformation, Vol. 1, World Sci. Publ, River Edge, NJ, 1997, pp. 313–400.

[25] B. Courcelle and J. Engelfriet, *Graph Structure and Monadic Second-Order Logic*, Cambridge University Press, 2012.

[26] A. Dawar, M. Grohe, and S. Kreutzer, *Locally excluding a minor*, in Proceedings of the 22nd IEEE Symposium on Logic in Computer Science (LICS 2007), Los Alamitos, CA, USA, 2007, IEEE, pp. 270–279.

[27] N. Derhy and C. Picouleau, *Finding induced trees*, Discrete Applied Mathematics, 157 (2009), pp. 3552–3557.

[28] F. V. Fomin, S. Gaspers, A. V. Pyatkin, and I. Razgon, *On the minimum feedback vertex set problem: Exact and enumeration algorithms*, Algorithmica, 52 (2008), pp. 293–307.

[29] F. V. Fomin, P. A. Golovach, and D. M. Thilikos, *Contraction obstructions for treewidth*, J. Comb. Theory, Ser. B, 101 (2011), pp. 302–314.

[30] F. V. Fomin, F. Grandoni, and D. Kratsch, *A measure & conquer approach for the analysis of exact algorithms*, J. ACM, 56 (2009).

[31] F. V. Fomin and D. Kratsch, *Exact Exponential Algorithms*, Springer, 2010.

[32] F. V. Fomin, D. Kratsch, I. Todinca, and Y. Villanger, *Exact algorithms for treewidth and minimum fill-in*, SIAM J. Comput., 38 (2008), pp. 1058–1079.

[33] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh, *Planar F-deletion: Approximation, kernelization and optimal FPT algorithms*, in Proceedings of 53rd Annual Symposium on Foundations of Computer Science (FOCS 2012), IEEE, 2012, pp. 470–479.

[34] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos, *Bidimensionality and kernels*, in Proceedings of the 21th ACM-SIAM Symposium on Discrete Algorithms (SODA 2010), SIAM, 2010, pp. 503–510.

[35] F. V. Fomin and Y. Villanger, *Finding induced subgraphs via minimal triangulations*, in Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS 2010), vol. 5 of Leibniz International Proceedings in Informatics, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2010, pp. 383–394.

[36] F. V. Fomin and Y. Villanger, *Subexponential parameterized algorithm for minimum fill-in*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2012, pp. 1737–1746.

[37] F. V. Fomin and Y. Villanger, *Treewidth computation and extremal combinatorics*, Combinatorica, 32 (2012), pp. 289–308.

[38] M. Frick and M. Grohe, *Deciding first-order properties of locally tree-decomposable structures*, J. ACM, 48 (2001), pp. 1184–1206.

[39] S. Gaspers, D. Kratsch, M. Liedloff, and I. Todinca, *Exponential time algorithms for the minimum dominating set problem on some graph classes*, ACM Transactions on Algorithms, 6 (2009).

[40] F. Gavril, *Algorithms on circular-arc graphs*, Networks, 4 (1974), pp. 357–369.

[41] F. Gavril, *Minimum weight feedback vertex sets in circle graphs*, Inform. Process. Lett., 107 (2008), pp. 1–6.

[42] ——, *Minimum weight feedback vertex sets in circle n-gon graphs and circle trapezoid graphs*, Discrete Math. Algorithms Appl., 3 (2011), pp. 323–336.

[43] P. A. Golovach, M. Kaminski, D. Paulusma, and D. M. Thilikos, *Induced packing of odd cycles in planar graphs*, Theor. Comput. Sci., 420 (2012), pp. 28–35.

[44] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[45] M. C. Golumbic and M. Lewenstein, *New results on induced matchings*, Discrete Applied Mathematics, 101 (2000), pp. 157–165.

[46] M. Grohe, *Logic, graphs, and algorithms*, in J.Flum, E.Grädel, T.Wilke (Eds), Logic and Automata-History and Perspectives, Amsterdam University Press, Amsterdam, 2007, pp. 357 – 422.

[47] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[48] S. Gupta, V. Raman, and S. Saurabh, *Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds*, SIAM J. Discrete Math., 26 (2012), pp. 1758–1780.

[49] P. Heggernes, *Minimal triangulations of graphs: A survey*, Discrete Mathematics, 306 (2006), pp. 297–317.

[50] T. Jian, *An $O(2^{0.304n})$ algorithm for solving maximum independent set problem*, IEEE Trans. Computers, 35 (1986), pp. 847–851.

[51] T. Kloks, D. Kratsch, and J. Spinrad, *On treewidth and minimum fill-in of asteroidal triple-free graphs*, Theor. Comput. Sci., 175 (1997), pp. 309–335.

[52] T. Kloks, D. Kratsch, and C. K. Wong, *Minimum fill-in on circle and circular-arc graphs*, J. Algorithms, 28 (1998), pp. 272–289.

[53] T. Kloks, C.-H. Liu, and S.-H. Poon, *Feedback vertex set on chordal bipartite graphs*, CoRR, abs/1104.3915 (2012).

[54] J. Kneis, A. Langer, and P. Rossmanith, *A fine-grained analysis of a simple independent set algorithm*, in IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009), R. Kannan and K. N. Kumar, eds., vol. 4 of Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2009, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 287–298.

[55] D. Kratsch, *The structure of Graphs and the Design of Efficient Algorithms*, habilitation thesis, F. Schiller Universität, Jena, 1996.

[56] S. Kreutzer, *Algorithmic meta-theorems*, in Finite and algorithmic model theory, vol. 379 of London Math. Soc. Lecture Note Ser., Cambridge Univ. Press, Cambridge, 2011, pp. 177–270.

[57] J. Lagergren and S. Arnborg, *Finding minimal forbidden minors using a finite congruence*, in Proceedings of the 8th International Colloquium on Automata, Languages and Programming (ICALP), vol. 510 of Lecture Notes in Computer Science, Springer, 1991, pp. 532–543.

[58] R. E. Miller and D. E. Muller, *A problem of maximum consistent subsets*, IBM Research Rep. RC-240, IBM T. J. Watson Research Center, New York, 1960.

[59] J. W. Moon and L. Moser, *On cliques in graphs*, Israel J. Math., 3 (1965), pp. 23–28.

[60] S. Parter, *The use of linear graphs in Gauss elimination*, SIAM Review, 3 (1961), pp. 119–130.

[61] M. Pergel, *Recognition of polygon-circle graphs and graphs of interval filaments is NP-complete*, in Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG), vol. 4769 of Lecture Notes in Computer Science, Springer, 2007, pp. 238–247.

[62] V. Raghavan and J. Spinrad, *Robust algorithms for restricted domains*, J. Algorithms, 48 (2003), pp. 160–172.

[63] N. Robertson and P. D. Seymour, *Graph minors. III. Planar tree-width*, J. Combin. Theory Ser. B, 36 (1984), pp. 49–64.

[64] ——, *Graph minors. V. Excluding a planar graph*, J. Combin. Theory Ser. B, 41 (1986), pp. 92–114.

[65] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, J. Combin. Theory Ser. B, 62 (1994), pp. 323–348.

[66] J. M. Robson, *Algorithms for maximum independent sets*, J. Algorithms, 7 (1986), pp. 425–440.

[67] D. J. Rose, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, New York, 1972, pp. 183–217.

[68] K. Suchan, *Minimal separators in intersection graphs*, Master's thesis, Akademia Gorniczo-Hutnicza im. Stanislawa Staszica w Krakowie, 2003.

[69] R. E. Tarjan and A. E. Trojanowski, *Finding a maximum independent set*, SIAM J. Computing, 6 (1977), pp. 537–546.

[70] C. Thomassen, *On the presence of disjoint subgraphs of a specified type*, J. Graph Theory, 12 (1988), pp. 101–111.

[71] F. van den Eijkhof, H. L. Bodlaender, and A. M. C. A. Koster, *Safe reduction rules for weighted treewidth*, Algorithmica, 47 (2007), pp. 139–158.

[72] M. Yannakakis, *The complexity of the partial order dimension problem*, SIAM Journal on Algebraic Discrete Methods, 3 (1982), pp. 351–358.

[73] M. Yannakakis and F. Gavril, *The maximum k-colorable subgraph problem for chordal graphs*, Inf. Process. Lett., 24 (1987), pp. 133–137.