SIAM J. COMPUT.
Vol. 44, No. 1, pp. 54–87

LARGE INDUCED SUBGRAPHS VIA TRIANGULATIONS AND CMSO*

FEDOR V. FOMIN†, IOAN TODINCA‡, AND YNGVE VILLANGER†

Abstract. We obtain an algorithmic metatheorem 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 the structure $(G[F], X)$ models φ, i.e., $(G[F], X) \models \varphi$. We give an algorithm solving this optimization problem on any 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. Pipelined with the known bounds on the number of potential maximal cliques in different graph classes, there are a plethora of algorithmic consequences extending and subsuming many known results on polynomial-time algorithms for graph classes. We also show that all potential maximal cliques of G can be enumerated in time $\mathcal{O}(1.7347^n)$. This implies the existence of an exact exponential algorithm of running time $\mathcal{O}(1.7347^n)$ for many NP-hard problems related to finding maximum induced subgraphs with different properties.

Key words. graph algorithms, treewidth, potential maximal cliques, minimal triangulations, CMSO

AMS subject classifications. 68R10, 05C85

DOI. 10.1137/140964801

1. Introduction.

We provide a generic algorithmic result concerning induced subgraphs with properties expressible in some logic. Our main algorithmic result 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 [58, 66]. The survey of Heggernes [49] gives an overview of techniques and applications of minimal triangulations. It was observed in the 1990s that minimal separators play an important role in obtaining minimal triangulations with certain properties. Techniques based on minimal separators were used to obtain polynomial algorithms computing the treewidth and minimum fill-in for different classes of graphs [10, 51, 50]. These results were extended by Bouchitté and Todinca in [12, 13], introducing 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 [33, 37]. Recently potential maximal clique based machinery was used to obtain a subexponential parameterized algorithm finding a minimum fill-in of a graph [38]. In the present paper

*Received by the editors April 14, 2014; accepted for publication (in revised form) December 1, 2014; published electronically February 12, 2015. Preliminary results contained in this paper were presented at STACS 2010 and SODA 2014.
http://www.siam.org/journals/sicomp/44-1/96480.html

†Department of Informatics, University of Bergen, Norway (fomin@ii.uib.no, yngve.villanger@uib.no). The first author was supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement 267959.

‡LIFO, Univ. Orléans, France (ioan.todinca@univ-orleans.fr). The author was partially supported by the ANR project AGAPE.

we provide completely new algorithmic applications of potential maximal cliques for problems of finding induced subgraphs with different properties.

Algorithmic applications of logic. Algorithmic metatheorems 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 metatheorems are based on deep relations between logic and combinatorial structures, which is a fundamental issue of computational complexity [46, 54]. A typical example of a metatheorem is the celebrated Courcelle's theorem [22] which states that all graph properties definable in monadic second order logic (MSO) can be decided in linear time on graphs of bounded treewidth. More recent examples of such metatheorems state that all first-order definable properties on planar graphs can be decided in linear time [39], 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 [27], and that all parameterized problems with finite integer index and additional "compactness" or "bidimensional" combinatorial properties, admit linear kernels on planar graphs [9, 35]. As it often happens with metatheorems, a combination of logic and graph theory not only gives a uniform explanation to tractability of many algorithmic problems but also provides 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 one is allowed to have sentences testing if a set is equal to q modulo r, for some integers q and r. Analogues and generalizations of Courcelle's theorem were obtained by Borie, Parker, and Tovey [11], Arnborg, Lagergren, and Seese [3], and Courcelle and Mosbah [25]. Our proof is using the framework of Borie, Parker, and Tovey [11].

1.1. Our results. A *property* $\mathcal{P}(G, X)$ on graphs associates with each graph G and each vertex subset X of G a boolean value. Borie, Parker, and Tovey [11] defined *regular properties*, whose definition we postpone till the next section. For all our applications, we need only the fact from Borie, Parker, and Tovey [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

$$\begin{array}{lll} & \text{Max} & |X| \\ (1.1) & \text{subject to} & \text{there is a set } F \subseteq V \text{ such that } X \subseteq F; \\ & & \text{the treewidth of } G[F] \text{ is at most } t; \\ & & (G[F], X) \models \varphi. \end{array}$$

For example, Maximum Independent Set can be encoded by (1.1) by taking $t = 0$, and φ expressing that $X = F$ and the absence of edges in $G[F]$. Similarly, Maximum Induced Forest[1] is encoded by taking $t = 1$, and φ expressing that $X = F$ and there is no cycle 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.

[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 the point of view of exact algorithms the two versions are equivalent, we choose to discuss the maximization problem.

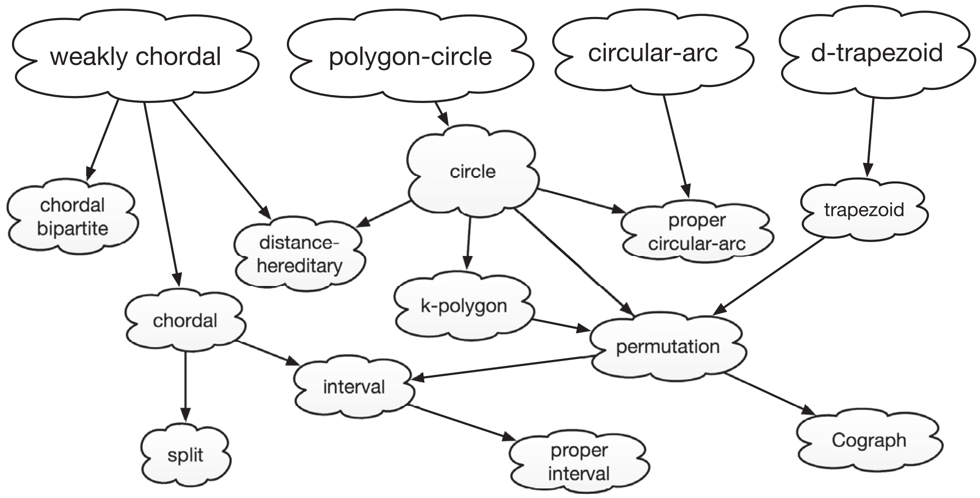

Fig. 1. *Graph classes with a polynomial number of potential maximal cliques.*

Let Π_G be the set of all potential maximal cliques in G. Our main result is that (1.1) 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 (1.1).

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.

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 Figure 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 (1.1) is solvable in polynomial time on such graph classes. We discuss in detail the bounds on the number of potential maximal cliques for different graph classes in section 6. Interestingly enough, while recognition of several of the 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 [60]. To the best of our knowledge, very few robust algorithms were known in the literature prior to our work.

Concerning exact exponential algorithms, most of the exact algorithms on maximum induced subgraph problems, like Maximum Independent Set or Maximum Induced Forest are so-called branching algorithms (a variation of Davis–Putnam-style exponential time backtracking [26]). In this work we make a step aside the "branching" path and use a completely new approach for problems related to finding induced subgraphs. It is worth noting that our approach is not only applicable to many problems where branching does not seem to work, but already for Maximum Induced Forest our algorithm provides a better running time than any of the previous (branching) algorithms. To demonstrate this, we give a new exponential algorithm enumerating potential maximal cliques. We prove the following theorem.

THEOREM 1.1. *The potential maximal cliques of an n-vertex graph can be listed in $\mathcal{O}(1.7347^n)$ time.*

Combined with Theorem 1.1, another direct consequence of our algorithm is that many intractable problems concerning maximum induced subgraphs with different properties expressible in the form of (1.1), can be solved in time $\mathcal{O}(1.7347^n)$. We are not aware of any such algorithmic metaresult in the area of exact algorithms showing that a wide range of problems can be solved significantly faster than by the trivial $\mathcal{O}(2^n)$-time brute-force algorithm. We remark that our framework not only captures a wide class of optimization problems, for most of the optimization problems including MAXIMUM INDUCED FOREST, it provides the fastest exponential algorithms known so far. Further implications are discussed in the next subsection.

1.2. Comparison with previous work.

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 the 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, Le, and Spinrad et al. [15] gives an extensive overview of different classes of graphs. By the seminal work of Grötschel, Lovász, and Schrijver [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, which can be seen as a natural extension of MAXIMUM INDEPENDENT SET, where instead of a 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 the 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, 52]. The containment relations between these classes of graphs is given in Figure 1.

According to the database at http://www.graphclasses.org on special graph classes the complexity of (weighted) MAXIMUM INDUCED FOREST on weakly chordal graphs 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 [16, 18, 19, 45]. Cameron and Hell in [17] 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 sets 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 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 [17] that for many graph classes including weakly chordal and polygon-circle graphs, $\mathcal{H}$-packing is solvable in polynomial time. Let us note that MAXIMUM INDUCED FOREST and $\mathcal{H}$-packing can be easily encoded as problem (1.1).

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 [32]. 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 $2^n n^{\mathcal{O}(1)}$ for every hereditary property π testable in polynomial time.

For the simplest property π, being edgeless, 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 [57] (see also Miller and Muller [56]) from the 1960s [68, 65, 31, 14, 70]. To the best of our knowledge, the fastest known algorithm of running time $\mathcal{O}(1.2002^n)$ is due to Xiao and Namagochi [70]. However, breaking the 2^n-barrier even for the case when π is being acyclic, i.e., of treewidth 1 or MAXIMUM INDUCED FOREST also known as MINIMUM FEEDBACK VERTEX SET, was an open problem in the area for some time. The first exact algorithm breaking the trivial 2^n-barrier is due to Razgon [61]. The running time $\mathcal{O}(1.8899^n)$ of the algorithm from [61] was improved in [29] to $\mathcal{O}(1.7548^n)$. Very recently, Xiao and Nagamochi [71] claimed an algorithm with running time $\mathcal{O}(1.7356^n)$. All these algorithms for MAXIMUM INDEPENDENT SET and MAXIMUM INDUCED FOREST are so-called branching algorithms (a variation of Davis–Putnam-style exponential-time backtracking [26]). There is also a relevant work of Gupta, Raman, and Saurabh [48] who gave algorithms for MAXIMUM INDUCED MATCHING and MAXIMUM 2-REGULAR INDUCED SUBGRAPH, with running times $\mathcal{O}(1.695733^n)$ and $\mathcal{O}(1.7069^n)$, respectively.

Potential maximal cliques. The notion of potential maximal cliques is due to Bouchitté and Todinca who used these objects to design polynomial-time algorithms computing treewidth and minimum fill-in of graphs with a polynomial number of minimal separators. Fomin et al. [33] have shown that the number of potential maximal cliques in an n-vertex graph is $\mathcal{O}(1.8135^n)$ and showed how to enumerate all potential maximal cliques in time $\mathcal{O}(1.8899^n)$. They also have shown how to compute the treewidth and the fill-in of a graph in time, up to polynomial of n, proportional to the time required to enumerate potential maximal cliques. The running time of the enumeration algorithm was improved by Fomin and Villanger in [37] to $\mathcal{O}(1.7549^n)$. Recall that in this article we further improve it to $\mathcal{O}(1.7347^n)$.

To sum up. Altogether we prove that the problems of the next table can be solved in $\mathcal{O}(1.7347^n)$ time for arbitrary graphs, and in polynomial time for all classes of graphs of Figure 1.

Problem	**Previous results**
Treewidth, Minimum Fill-In	$\mathcal{O}(1.7549^n)$ [37]
Maximum Induced Forest=Minimum Feedback Vertex Set	$\mathcal{O}(1.7356^n)$ [71]
Independent $\mathcal{H}$-packing	
Maximum Induced Subgraph excluding a planar minor	
k-in-a-Path, k-in-a-Tree	
Maximum Induced Matching	$\mathcal{O}(1.6958^n)$ [48]
Maximum Independent Set	$\mathcal{O}(1.2002^n)$ [70]

Further applications are discussed in section 5 for arbitrary graphs, and in section 6 for graph classes.

The remaining part of this paper is organized as follows. Section 2 provides definitions and preliminary results. In section 3, we provide in detail our main result, the algorithm solving problem (1.1), i.e., computing an optimal induced subgraph for regular property $\mathcal{P}$ and of treewidth at most t. Section 4 contains the improved algorithm enumerating potential maximal cliques. In section 5, we discuss different applications of our main results and show how various problems can be expressed in the form of the main optimization problem. For some special graph classes the optimization problem (1.1) can be used to capture even more problems; how to do it is discussed in section 6. We conclude in section 7 with open problems.

2. Preliminaries. We denote by $G = (V, E)$ a finite, undirected, and simple graph with $|V| = n$ vertices and $|E| = m$ edges. We also use $V(G)$ for the vertex set of G and $E(G)$ for its edge set. For a vertex set $S \subseteq V$, we use $G[S]$ to denote the subgraph of G induced by S, and $G - S$ denotes the graph $G[V \setminus S]$. 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 $\bigcup_{v\in S} N(v) \setminus S$. We say that a vertex set C is *connected*, if it induces a connected subgraph. A *connected component* of a graph G is a maximal connected vertex subset.

The notion of treewidth is due to Robertson and Seymour [62]. 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 CMSO, an extension of MSO, as a basic tool to express properties of vertex/edge sets in graphs.

The syntax of 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. **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 MSO, 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 CMSO. So essentially CMSO is MSO with the following atomic sentence for a set S:

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

We refer to [3, 21, 23] and the book of Courcelle and Engelfriet [24] for a detailed introduction to CMSO. In [24], CMSO is referred to as CMS_2.

2.1. *t*-terminal recursive graphs and regular properties. We also 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 nonzero 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 2.1 (see [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 subexpression. 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 with each graph G and each vertex subset X of G a boolean value. By the celebrated results of [21, 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 subexpression and the partial solutions (i.e., possible subsets of X) encountered so far. Let G_1 be such a subexpression 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, Parker, and Tovey [11].

Let $G = (V, T, E)$ be a $(t+1)$-terminal recursive graph. For a 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 2.2 (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 the following:*

- *(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 information, 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 if and only if $G[X]$ is 3-colorable. We show that it is regular. Let $P_3(t)$ be the set of partitions of subsets of $\{1, 2 \ldots, t+1\}$ 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 nonaccepting 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 2.3 (Borie, Parker, and Tovey [11]). *Any property $\mathcal{P}(G, X)$ expressible by a CMSO formula is regular.*

Moreover, the result of Borie, Parker, and Tovey [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 formulas, which allows an arbitrary number of free variables over vertices, edges, vertex sets, and edge sets. For our applications, it is sufficient to consider properties over graphs and one vertex set, corresponding to formulas 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. Minimal triangulations and potential maximal cliques.

Chordal graphs and triangulations. 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.

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.

We will use the following relation between treewidth and triangulations.

PROPOSITION 2.4 (Folklore). *For any graph G, $\mathrm{tw}(G) \le k$ if and only if there is a triangulation H of G with the maximum clique size at most $k+1$.*

Let u and v be two non adjacent vertices of a graph G. Given a vertex subset S, recall that $G-S$ denotes the graph $G[V(G) \setminus S]$. A set of vertices $S \subseteq V$ is a u,v-*separator* if u and v are in different connected components of the graph $G-S$. A connected component $G[C]$ of $G-S$ is a *full component associated with* 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. The following proposition is an exercise from the book of Golumbic [44].

PROPOSITION 2.5 (folklore). *Let $G=(V,E)$ be a graph and $S \subseteq V$ be a vertex subset. Then S is a minimal separator of G if and only if there are two components C and D of $G-S$ such that $N(C)=N(D)=S$.*

We will need the following result of Berry, Bordat, and Cogis [5].

PROPOSITION 2.6 (see [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.

Let us give their main characterization obtained by Bouchitté and Todinca [12].

PROPOSITION 2.7 (see [12]). *Let $\Omega \subseteq V$ be a set of vertices of the graph $G=(V,E)$ and $\{C_1,\dots,C_p\}$ be the set of connected components of $G-\Omega$. We define $\mathcal{S}(\Omega)=\{S_1,S_2,\dots,S_p\}$, where $S_i=N(C_i)$, $i\in\{1,2,\dots,p\}$, is the set of those vertices of Ω which are adjacent to at least one vertex of the component C_i. Then Ω is a potential maximal clique of G if and only if*

1. *each $S_i \in \mathcal{S}(\Omega)$ is strictly contained in Ω;*
2. *the graph on the vertex set Ω obtained from $G[\Omega]$ by completing each $S_i \in \mathcal{S}(\Omega)$ into a clique is a complete graph.*

Moreover, if Ω is a potential maximal clique, then $\mathcal{S}(\Omega)$ is the set of minimal separators of G contained in Ω.

For example, when G is a cycle, the minimal separators of G are exactly the pairs of nonadjacent vertices, and the potential maximal cliques are exactly the triples of vertices. Let us note that by Proposition 2.7, for every pair of nonadjacent vertices $x,y\in\Omega$ there is a connected component C of $G-\Omega$ such that $x,y\in N(C)$.

We will use the following result from [13].

PROPOSITION 2.8 (see [13]). *Let Π_G denote the set of all potential maximal cliques of graph G. Then $|\Pi_G| \le n|\Delta_G|^2+n|\Delta_G|+1$, and the set Π_G can be listed in time $\mathcal{O}(n^2m|\Delta_G|^2)$.*

Let us note that by Proposition 2.8, if graphs from some class have a polynomial number of minimal separators, these graphs also have a polynomial number of potential maximal cliques.

Let Ω be a potential maximal clique. By Proposition 2.7, a subset $S\subseteq\Omega$ is a minimal separator of G if and only if S is the neighborhood of a connected component of $G-\Omega$. For a minimal separator S and a full connected component C of $G-S$, we say that (S,C) is a *full block* associated with 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 [33], the number of good triples is at most $n|\Pi_G|$.

Let F be a vertex subset of graph G and let T_F be a triangulation of the subgraph $G[F]$ induced by F. We say that a minimal triangulation T_G of G *respects* T_F if for every clique K of T_G, its intersection with F is either empty, or is a clique in T_F. The following lemma about the existence of respecting minimal triangulations is the crucial part of our main algorithm.

LEMMA 2.9 (respecting triangulation lemma). *Let F be a vertex set of graph G and let T_F be a minimal triangulation of the subgraph $G[F]$ induced by F. Then there exists a minimal triangulation T_G of G which respects T_F.*

Proof. Let $T_F = (F, E_F)$ be a minimal triangulation of the subgraph $G[F]$ induced by F. We show that there exists a minimal triangulation T_G of such that T_F is an induced subgraph of T_G. This would imply that T_G respects T_F.

First construct a graph $H = (V, E')$ with the same vertex set as $G = (V, E)$, such that $H[F] = T_F$ and each vertex of $V \setminus F$ is adjacent to all other vertices of H. Therefore H is a supergraph of G, in which $V \setminus F$ induces a clique and such that all edges are present between F and $V \setminus F$. Let us first prove that H is a triangulation of G, i.e., that H is chordal. By contradiction, assume there is a set of at least four vertices $C \subseteq V$ such that $H[C]$ is a chordless cycle. Observe that C cannot be contained in F because $H[F] = T_F$ is chordal. Let x be a vertex of $C \setminus F$. By construction of H, vertex x is adjacent to all other vertices of C, hence x is of degree at least three in $H[C]$. This contradicts the assumption that $H[C]$ is a chordless cycle.

Now the triangulation $H = (V, E')$ of $G = (V, E)$ contains a minimal triangulation $T_G = (V, E'')$ of G such that $E \subseteq E'' \subseteq E'$. It remains to prove that $T_G[F] = T_F$. Since $T_G[F]$ is chordal, this graph is a triangulation of $G[F]$. Moreover, by construction $T_G[F]$ is a subgraph of T_F. By the fact that T_F is a *minimal* triangulation of $G[F]$ we conclude that $T_G[F] = T_F$. ∎

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 5, 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 $\mathcal{F}$ [63]. 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 [17]. 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 are 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 3.1. *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 2.3, results of Theorem 3.1 hold for every property $\mathcal{P}(G, X)$ expressible by a CMSO formula. Combined with Propositions 2.8 and Theorem 1.1, we obtain the following application of Theorem 3.1.

Corollary 3.2. *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 a polynomial number of minimal separators.*

3.1. Notation and data structures. Our algorithm proceeds by dynamic programming on blocks and on good triples. 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 2.2.

Let F be a vertex subset of vertices of G. By Proposition 2.4, if $G[F]$ is of treewidth at most t, then there exists a (minimal) triangulation T_F of $G[F]$ of width at most t, and by Lemma 2.9, there is a minimal triangulation T_G of G respecting T_F.

In the algorithm we compute partial compatible solutions by dynamic programming on blocks and good triples. The next definition and the following notation are crucial for our algorithm.

Definition 3.3 (partial compatible solution). *Let (S, C) be a full block and (S, C, Ω) be a good triple of graph G. 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 $\mathcal{P}$. Let also $X \subseteq F \subseteq V(G)$. 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 T_F of $G[F]$ of width at most t and a minimal triangulation T_G of G respecting T_F, such that S is a minimal separator (resp., Ω is a maximal clique) of T_G.*

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

The definitions are illustrated in Figure 2. For simplicity, the subset X of F is not depicted on the figures.

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 $-\infty$.

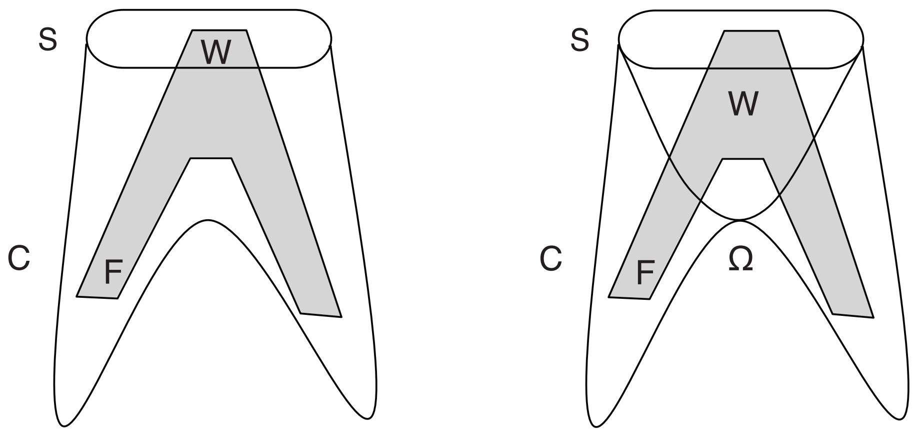

Fig. 2. *Partial solutions compatible with* (S, C, W, c) *(left), and with* (S, C, Ω, W, c) *(right). Set* F *is depicted in gray. Note that set* W *corresponds to* $F \cap S$ *in the first case, and to* $F \cap \Omega$ *in the second case.*

3.2. The algorithm. Our algorithm proceeds by dynamic programming on full blocks and good triples. By [33], the number of good triples is $\mathcal{O}(n|\Pi_G|)$. The blocks are first sorted by size in an increasing order. 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 its potential maximal cliques 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 subject to these constraints, X is of maximum size

Order all full blocks by inclusion;
for *all full blocks* (S, C) *in this order* **do**
  for *all 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 (3.1);
    else
      Compute $\beta(S, C, \Omega, W, c)$ using (3.3), (3.4), (3.5), and (3.6);
  for *all* $W \subseteq S$ *of size* $\leq t + 1$ *and all* $c \in \mathcal{C}$ **do**
    Compute $\alpha(S, C, W, c)$ using (3.2);
Compute an optimal solution using (3.7);

When computing partial solutions $(G[F], X)$ compatible with a quadruple (S, C, W, c) or with a quintuple (S, C, Ω, W, c) we need to compute, from the class c and the ordered set of terminals W, the intersection $X \cap W$. Consider a $(t + 1)$-terminal recursive graph $D = (V_D, T, E_D)$ and let c be a homomorphism class for our property $\mathcal{P}$. Although this is not explicitly required by the definition of regular properties (Definition 2.2), we may assume without loss of generality 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

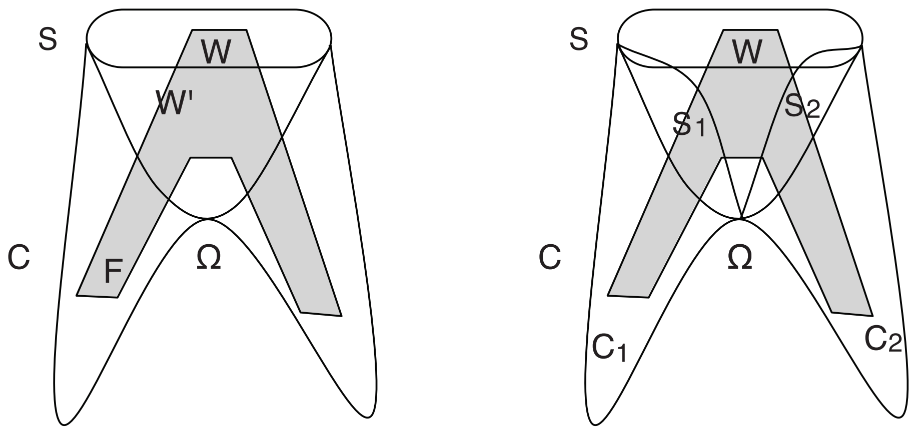

FIG. 3. *Computing α form β (left), and β from α (right).*

T and such a vertex subset Y.) Hence 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 every Y such that $h(D, Y) = c$.

The base case. In the base case we have a minimal full block (S, C). In this case (see, e.g., [12]), we have that (S, C, Ω) is a good triple for $\Omega = S \cup C$. 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). Hence for the base case we have

$$\beta(S, C, \Omega, W, c) = \begin{cases} |X| & \text{if there is } X \subseteq W \text{ such that } h(G[W], X) = c, \\ -\infty & \text{otherwise.} \end{cases} \tag{3.1}$$

The computation of each value $\beta(S, C, \Omega, W, c)$ corresponding to every base case takes $\mathcal{O}(n)$ time, because we have to store the value in a table indexed by (S, C, Ω, c). For each good triple, we try at most n^{t+1} sets W. The number of good triples is $\mathcal{O}(n|\Pi_G|)$, so altogether these computations take $\mathcal{O}(n^{t+3}|\Pi_G|)$ 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); the situation is depicted in Figure 3 (left). Let T_F be a triangulation of $G[F]$ as in Definition 3.3 and let T_G be a minimal triangulation of G respecting T_F. Let Ω be the maximal clique of T_G 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 $W' \setminus W$ from the set of terminals, and possibly renumbering the remaining terminals. Therefore $\odot_{f(W',W)}(c') = c$.

We claim the following.

LEMMA 3.4.

$$\alpha(S, C, W, c) = \max \beta(S, C, \Omega, W', c'), \tag{3.2}$$

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 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 $\mathcal{O}(n)$ 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 Proposition 2.7, (S_i, C_i) are full blocks; see also [12], and Figure 3 (right) for an illustration.

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 the set of terminals and with $E_i^+ = E(G[F_i \cup W])$ as the 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 the 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 $(t+1)$-terminal recursive graph having W as the set of terminals, and c_W be the homomorphism class $h(G[W], X_W)$.

LEMMA 3.5.

$$\delta_i(S, C, \Omega, W, c_i^+) = \max \alpha(S_i, C_i, W_i, c_i) + |term(c_W, W) \setminus term(c_i, W_i)|, \tag{3.3}$$

where the maximum is taken 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 $F_i = F_i^+ \cap (S_i \cup C_i)$, $X_i = X_i^+ \cap (S_i \cup C_i)$, and $X_W = X \cap W$. 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 in $\Omega \cup C_1 \cup \cdots \cup C_i$. It corresponds to the gluing of some partial solutions $(H_1^+, X_1^+), \ldots, (H_i^+, X_i^+)$.

LEMMA 3.6. *Function γ_i is computed as follows. For $i = 1$,*

$$\gamma_1(S, C, \Omega, W, c) = \delta_1(S, \Omega, C, W, c). \tag{3.4}$$

For each $i \in \{2, \ldots, p\}$,

$$\gamma_i(S, C, \Omega, W, c) = \max \gamma_{i-1}(S, C, \Omega, W, c') + \delta_i(S, \Omega, C, W, c'') - |term(c', W)|, \tag{3.5}$$

where the maximum is taken 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 $(G[F \setminus C_i], X \setminus C_i)$ is a partial solution compatible with (S, Ω, C, W, c'') and $(G[F \setminus (C_1 \cup \cdots \cup C_{i-1})], X \setminus (C_1 \cup \cdots \cup C_{i-1})))$ is a partial solution compatible with (S, C, Ω, W, c') for some classes c' and c'' such that $\odot_{g(W)}(c', c'') = c$. The term $|term(c', W)|$ corresponds to $X \cap W$ and avoids overcounting of these vertices. □

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

LEMMA 3.7.

$$\beta(S, C, \Omega, W, c) = \gamma_p(S, C, \Omega, W, c). \tag{3.6}$$

We claim that for a fixed quadruple (S, C, Ω, W) computing the values $\beta(S, C, \Omega, W, c)$ from values α, takes $\mathcal{O}(n^2)$ time. Again by [33], the smaller blocks (S_i, C_i) can be listed in $\mathcal{O}(m)$ time. For each i, the computation of function $\delta_i(S, \Omega, C, W, c_i^+)$ takes $\mathcal{O}(|S_i| + |C_i|) = \mathcal{O}(n)$ time, because we need to access the values $\alpha(S_i, W_i, C_i, c_i)$. Computing $\gamma_i(S, C, \Omega, W, c)$ from values γ_{i-1} and δ_i can be done in $\mathcal{O}(n)$ time for each i.

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

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

LEMMA 3.8. *The solution size is*

$$\max \alpha(\emptyset, V, \emptyset, c), \tag{3.7}$$

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

Proof. By the 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 Lemma 2.9, this solution is compatible with $\alpha(\emptyset, V, \emptyset, c)$ for the class c of the $(t+1)$-terminal graph $(F, \emptyset, E(G[F]))$, which completes the proof of the lemma. ◻

This latter computation takes constant time.

The total running time of the algorithm is $\mathcal{O}(|\Pi_G| n^{t+4})$. 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 3.1 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 $w : V \to \mathbb{Z}$, 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.

Observe that the weight function gives an *integer* weight to each vertex, thus arithmetic operations can be performed in polynomial time in the input size.

Theorem 3.9. *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* $|\Pi_G| n^{\mathcal{O}(1)} \log M$ *time, where M is the maximum $|w(v)|$, $v \in V(G)$, and when Π_G is given in the input.*

In particular the problem can be solved in $\mathcal{O}(1.7347^n \log M)$ time for arbitrary graphs, and in polynomial time for classes of graphs with a 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 $U \cap S \not\subseteq W$ (resp., $U \cap \Omega \not\subseteq W$) is immediately set to $-\infty$, meaning that such a partial solution is rejected.

In order to maximize the weight of the solution, the values $\alpha(S, C, W, c)$ (resp., $\beta(S, C, \Omega, W, c)$) 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 (3.1), $|term(c', W)|$ in (3.5), and $|term(c_W, W) \setminus term(c_i, W_i)|$ in (3.3)) by the weights of these sets.

We also point out that the weights can be negative. In particular, we can use Theorem 3.9 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 3.1 and 3.9. 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 that the size of X' is exactly v'. The computation of α and β is quite straightforward, by adapting (3.1)–(3.7). 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, Parker, and Tovey [11] proved that *all* properties expressible by CMSO formulas 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 (3.1)–(3.7). 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 gluing 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 the maximum induced subgraph of treewidth at most t, the one with minimum dominating set.

4. Listing potential maximal cliques. This section is devoted to the proof of Theorem 1.1. Our enumeration algorithm is based on the characterization of potential maximal cliques from Proposition 2.7 and the following "firefighters lemma."

Proposition 4.1 (see [37]). *Let $G = (V, E)$ be a graph. For every vertex v and a pair of integers $b, f \geq 0$, the number of connected vertex subsets $B \subseteq V$ such that*

- $v \in B$,
- $|B| = b + 1$, *and*
- $|N(B)| = f$

is at most $\binom{b+f}{f}$. Moreover, all these sets B can be listed in time $\mathcal{O}^*(\binom{b+f}{f})$.

The enumeration of all potential maximal cliques is provided by Algorithm 2. We start with a brief overview of this algorithm. The algorithm distinguishes five types of potential maximal cliques. We shall prove that each of these types is listed within the desired running time, and that altogether the five types cover all possible potential maximal cliques. When we say that a potential maximal clique is "of the ith type" we implicitly mean that it is not of the jth type for all $j < i$.

Let Ω be a potential maximal clique and let $x \in \Omega$. We denote by D_x the union of the vertex sets of all connected components $G[C]$ of $G - \Omega$ such that $x \in N(C)$.

ALGORITHM 2. ENUMERATE ALL POTENTIAL MAXIMAL CLIQUES.

// Part (1), first type;
for *each edge* $xy \in E$ **do**
  let G_{xy} be graph $G - xy$;
  for *each minimal* x, y*-separator* S *of* G_{xy} **do**
    if $S \cup \{x, y\}$ *is a potential maximal clique of* G **then**
      output $S \cup \{x, y\}$;

// Part (2), second type: α is a constant s.t. $0 < \alpha < 1$, it will be set to 0.239;
for *each vertex set* $Z \subset V$ *such that* $|Z| \leq \alpha n$ **do**
  if $N(Z)$ *is a potential maximal clique* **then**
    output $N(Z)$;

// Part (3), third type: β is a constant s.t. $0 \leq \beta \leq 1$ and $\alpha + \beta > 1$; it will be set to 0.794;
for *each vertex* z **do**
  for *each connected vertex set* Z *such that* $z \in Z$ *and* $|N[Z]| \leq \beta n$ **do**
    if $N(Z) \cup \{z\}$ *is a potential maximal clique* **then**
      output $N(Z) \cup \{z\}$;

// Part (4), fourth type: γ is a constant depending on α and β, it will be 7;
for *each vertex set* Z *of size at most* $3/5(n - |N(Z)|)$ **do**
  if $G[Z]$ *has at most* γ *connected components* ***and*** $N(Z)$ *is a potential maximal clique* **then**
    output $N(Z)$;

// Part (5), fifth type;
for *all possible values* $c_1, c_2, c_3, \bar{n}_1, \bar{n}_2, \bar{n}_3, a$ *such that* $c_1 + c_2 + c_3 + \bar{n}_1 + \bar{n}_2 + \bar{n}_3 + a \leq n$ **do**
  select $i, j, k \in \{1, 2, 3\}$, $i \neq j \neq k$, minimizing the value of $\binom{c_i + a + \bar{n}_j + \bar{n}_k}{c_i}\binom{c_j + \bar{n}_i}{c_j}$;
  for *each connected vertex set* C_i *such that* $|C_i| = c_i$ *and* $|N(C_i)| = \bar{n}_j + \bar{n}_k + a$ **do**
    for *each connected vertex set* C_j *such that* $|C_j| = c_j$ *and* $|N(C_j) \setminus N(C_i)| = \bar{n}_i$ **do**
      if $N(C_i \cup C_j)$ *is a potential maximal clique* **then**
        output $N(C_i \cup C_j)$;

We say that a potential maximal clique Ω is of the *first type* if for some $x \in \Omega$, set $N(D_x)$ is strictly contained in Ω. In this case we simply guess x and a vertex y in Ω but not in $N(D_x)$, and show in Claim 1 that $\Omega - \{x, y\}$ is a minimal separator in graph G_{xy} obtained from G by removing edge xy. Thus the enumeration of potential maximal cliques of the first type boils down to enumeration of minimal separators in graph G_{xy}.

Therefore, for each potential maximal clique Ω that is not of the first type, we have that for every $x \in \Omega$, $\Omega = N(D_x)$. Let now $\hat{D}_x$ denote the union of vertex sets of *some* connected components $G[C]$ of $G - \Omega$ such that $\Omega = N(\hat{D}_x)$ and $\hat{D}_x$ is minimal under these constraints.

We say that a potential maximal Ω is of the *second type* if $\hat{D}_x$ is "small" for some $x \in \Omega$. By "small" we mean that $|\hat{D}_x| \leq \alpha n$ for a constant α, that later will be set to 0.239. In this case we enumerate sets $\hat{D}_x$ by brute force and for each compute $\Omega = N(\hat{D}_x)$.

The *third type* corresponds to potential maximal cliques Ω that for some $x \in \Omega$, set $D_x \cup \Omega$ is "relatively small," i.e. of size at most βn for some constant β, that will be set to 0.794. In this case we shall see that $\Omega \setminus \{x\} = N(D_x \cup \{x\})$. Since $D_x \cup \{x\}$ is connected, we will be able to use the firefighters lemma (Proposition 4.1) to enumerate potential maximal cliques of this type.

We then prove (Claim 4) that for every potential maximal clique Ω that is not of the first three types, for every $x \in \Omega$, the vertex set $\hat{D}_x$ is the union of vertex sets of a constant number γ of connected components of $G - \Omega$. Constant γ depends on constants α and β, in our case it will be equal to 7.

For a potential maximal clique Ω of the *fourth type*, we have that for some $x \in \Omega$, there is a subset $\hat{D}'_x$ of $\hat{D}_x$ such that $N(\hat{D}'_x) = \Omega$, $G[\hat{D}'_x]$ has at most γ connected components, and $|\hat{D}'_x| \leq 3/5(n - |\Omega|)$. We use again the firefighters lemma to control the running time in this case.

All other potential maximal cliques form the *fifth type*. We prove in Claim 7 that in this case in graph $G - \Omega$ it is possible to select three connected components such that for each two of them, the neighborhood of their union is exactly Ω. To enumerate potential maximal cliques of the fifth type, we combine the firefighters lemma and several observations on the possible sizes of these components and their neighborhoods.

We prove through a sequence of steps that our algorithm lists all potential maximal cliques of G and after that we analyze its running time. Recall that, given a potential maximal clique Ω and a vertex $x \in \Omega$, D_x denotes the union of the vertex sets of all connected components of $G - \Omega$ "seeing" x.

LEMMA 4.2 (see also [33]). *Let Ω be a potential maximal clique of $G = (V, E)$, and let $x \in \Omega$. Then $N(D_x \cup \{x\}) = \Omega \setminus \{x\}$.*

Proof. By Proposition 2.7, for every $y \in \Omega \setminus \{x\}$, we have that x and y are either adjacent, or they are both in the neighborhood $N(C)$ of some connected component C of $G - \Omega$. In the first case $y \in N(x)$, in the second case $y \in N(D_x)$. Recall that $N(D_x) \subseteq \Omega$ and $N(x) \subseteq D_x \cup \Omega$ by the definition of D_x. Thus $N(D_x \cup \{x\}) = \Omega \setminus \{x\}$. ∎

CLAIM 1. *Let Ω be a potential maximal clique of the first type, i.e., there exist $x, y \in \Omega$ such that $y \notin D_x$. Then $\Omega \setminus \{x, y\}$ is an x, y-minimal separator in the graph $G_{xy} = G - xy$ obtained by removing edge xy from G.*

Proof. Observe that $x \notin N(D_y)$. By Lemma 4.2, we have that $N_G(D_x \cup \{x\}) = \Omega \setminus \{x\}$ and $N_G(D_y \cup \{y\}) = \Omega \setminus \{y\}$. Thus, in the graph G_{xy}, the vertex sets $D_x \cup \{x\}$ and $D_y \cup \{y\}$ are disjoint and connected (as in G) and their neighborhood is $\Omega \setminus \{x, y\}$. By Proposition 2.5, $\Omega \setminus \{x, y\}$ is an x, y-minimal separator of G_{xy}. ∎

Therefore Part (1) of Algorithm 2 enumerates all potential maximal cliques of the first type. From now on we assume that potential maximal clique Ω is not of the first type; hence, $N(D_x) = \Omega$ for all $x \in \Omega$. We define vertex set $\hat{D}_x \subseteq D_x$ as the union of an inclusion minimal set of connected component of $G - \Omega$ seeing Ω. In other words, we define $\hat{D}_x = \bigcup_{i=1}^{r} C_i$ such that

- for each $i \in \{1, \ldots, r\}$, C_i is a connected component of $G - \Omega$;
- $N(\hat{D}_x) = \Omega$;
- for each $i \in \{1, \ldots, r\}$, $N(\hat{D}_x \setminus C_i) \neq \Omega$.

CLAIM 2. *Let Ω be a potential maximal clique of the second type, i.e., such that $|\hat{D}_x| \leq \alpha n$ for some $x \in \Omega$. Then Ω is listed in Part (2) of Algorithm 2.*

Proof. Part (2) of Algorithm 2 tests each vertex set Z of size at most αn and outputs the set $N(Z)$ if $N(Z)$ is a potential maximal clique. Thus every potential maximal clique of the second type is listed when $Z = \hat{D}_x$. ∎

The following claim follows directly from Lemma 4.2.

CLAIM 3. *Let Ω be a potential maximal clique of the third type, in particular $N[D_x \cup \{x\}] \le \beta n$ for some $x \in \Omega$. Then Ω is listed by Part* (3) *of Algorithm* 2.

The next claim concerns connected components of potential maximal cliques not listed during the first three steps of the algorithm.

CLAIM 4. *Let Ω be a potential maximal clique that is not of the first three types. Let $\gamma = 1 + \frac{1-\beta}{\alpha+\beta-1}$. Then for each vertex $x \in \Omega$, graph $G[\hat{D}_x]$ contains at most γ connected components.*

Proof. Let $C_1, C_2, \ldots, C_r$ be the connected components of $G[\hat{D}_x]$ given in the ascending order according to their sizes. We first prove that $|\hat{D}_x \setminus C_1| \le n - \beta n$. By the minimality of $\hat{D}_x$, there is $z \in N(C_1)$ that does not belong to any other $N(C_i)$ for $i \ge 2$. Therefore $N[D_z] \cap (\hat{D}_x \setminus C_1) = \emptyset$. In particular, if $|\hat{D}_x \setminus C_1| > n - \beta n$ then $N[D_z] \le \beta n$, contradicting the fact that Ω is not of the third type. Because Ω is not of the second type, we have that $|\hat{D}_x| > \alpha n$. We deduce that $|C_1| \ge (\alpha + \beta - 1)n$. Since $|C_1| \ge (\alpha+\beta-1)n$ and $|\hat{D}_x \setminus C_1| \le n - \beta n$, we have that $r - 1 \le \frac{1-\beta}{\alpha+\beta-1}$. Thus $r \le \gamma$. ∎

From now on we focus on potential maximal cliques Ω that are not of the first three types. By Proposition 2.7, for every pair of nonadjacent vertices $x, y \in \Omega$ there is a connected component C of $G - \Omega$ such that $x, y \in N(C)$. Because Ω is not of the first type, we have that for every $y \in \Omega$, there is a connected component C of $G[\hat{D}_x]$ such that $x, y \in N(C)$. Let

$$\mathcal{C} = \bigcup_{x \in \Omega} \hat{D}_x.$$

The next claim shows that the connected components of $G[\mathcal{C}]$ are minimal subject to pairs of nonadjacent vertices of Ω.

CLAIM 5. *For every connected component C of graph $G[\mathcal{C}]$, there is a "private" pair of nonadjacent vertices $x, y \in \Omega$, i.e., a pair such that*

- *$x, y \in N(C)$, and*
- *there is no connected component $C' \neq C$ of $G[\mathcal{C}]$ such that $x, y \in N(C')$.*

Proof. Targeting towards a contradiction, suppose there is a connected component C of graph $G[\mathcal{C}]$ without a private pair. By the minimality of each of the sets $\hat{D}_x$, $x \in \Omega$, the only reason why C contributes to $G[\mathcal{C}]$ is that there is $y \in \Omega$, such that C is the connected component of $G[\hat{D}_y]$ and $N(\hat{D}_y \setminus C) \neq \Omega$. Let $z \in \Omega \setminus N(\hat{D}_y \setminus C)$. Because C has no private pair, there is a connected component C' of $G[\mathcal{C}]$ such that $y, z \in N(C')$. Thus C' is also a connected component of $G[\hat{D}_y]$ and hence $z \notin \Omega \setminus N(\hat{D}_y \setminus C)$, which is a contradiction. ∎

CLAIM 6. *For every connected component C of $G[\mathcal{C}]$, there are two vertices x and y such that $C = \hat{D}_x \cap \hat{D}_y$.*

Proof. By Claim 5, C has a private pair $x, y \in \Omega \cap N(C)$ such that for every other connected component C' of $G[\mathcal{C}]$, we have that at least one of x or y is not contained in $N(C')$. Thus $C = \hat{D}_x \cap \hat{D}_y$. ∎

CLAIM 7. *Let Ω be a potential maximal clique not of the first three types. Then*

1. *either there exists $x \in \Omega$ such that $|\hat{D}_x| \le 3/5(n - |\Omega|)$,*
2. *or $G[\mathcal{C}]$ contains at most* 3 *connected components, and for every pair of such components C_i and C_j, we have that $N(C_i \cup C_j) = \Omega$.*

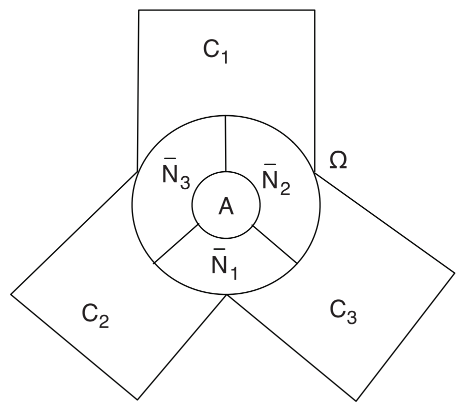

FIG. 4. *Potential maximal cliques of the fifth type.*

Proof. Let $C_1, C_2, \ldots, C_r$ be the connected components of $G[\mathcal{C}]$, and let us assume that C_1 is of minimum size. By Claim 6 applied to C_1, there exist $x, y \in \Omega$ such that $C_1 = \hat{D}_x \cap \hat{D}_y$.

Consider first the case when $|C_1| \leq (1/5)(n - |\Omega|)$. Then the size of one of the sets $\hat{D}_x$, $\hat{D}_y$ does not exceed

$$|C_1| + \frac{n - |\Omega| - |C_1|}{2} = \frac{n - |\Omega| + |C_1|}{2},$$

which is at most $(3/5)(n - |\Omega|)$ for $|C_1| \leq (1/5)(n - |\Omega|)$.

For the case $|C_1| > (1/5)(n - |\Omega|)$, because C_1 is the connected component of the smallest size, it follows that $r \leq 4$. Towards the contradiction, let us assume that $r = 4$. Observe that each of $\hat{D}_x$ and $\hat{D}_y$ is formed by the union of at least two sets C_i. Indeed if $\hat{D}_x = C_1$ then $N_G(C_1) = \Omega$, contradicting Proposition 2.7. Because $C_1 = \hat{D}_x \cap \hat{D}_y$, one of the sets $\hat{D}_x$, $\hat{D}_y$ contains exactly two components. Say, $\hat{D}_x = C_1 \cup C_2$. If $|C_1 \cup C_2| \leq (3/5)(n - |\Omega|)$ then the first condition of the claim holds. If $|C_1 \cup C_2| > (3/5)(n - |\Omega|)$, then at least one of C_3 and C_4 should be of size at most $(1/5)(n - |\Omega|)$, and thus of size smaller than $|C_1|$ which is a contradiction. Thus $r \leq 3$.

Also $r > 2$, because each of the sets $\hat{D}_x$ and $\hat{D}_y$ contains at least two connected components and $C_1 = \hat{D}_x \cap \hat{D}_y$. Thus $G[\mathcal{C}]$ contains exactly three components C_1, C_2, and C_3. Thus $\hat{D}_x$ contains exactly two of them, say C_1 and C_2, and $\hat{D}_x$ contains C_1 and C_3. Then $N(C_1 \cup C_2) = N(\hat{D}_x) = \Omega$ and $N(C_1 \cup C_3) = N(\hat{D}_y) = \Omega$. It remains to prove that $N(C_2 \cup C_3) = \Omega$. For component C_2 there is a private pair $u, v \in \Omega$ such that $C_2 = \hat{D}_u \cap \hat{D}_v$. Then either $\hat{D}_u$ or $\hat{D}_v$ is equal to $C_2 \cup C_3$, and thus $N(C_2 \cup C_3) = \Omega$. ∎

All potential maximal cliques of the fourth type (corresponding to the first item of the previous claim) are listed by Part (4) of Algorithm 2. It remains to prove that the fifth case of the algorithm enumerates all remaining potential maximal cliques.

CLAIM 8. *Let Ω be a potential maximal clique that is not of one of the first four types. Then Ω is listed by Part* (5) *of Algorithm* 2.

Proof. By Claim 7, there are three connected components C_1, C_2, C_3 of $G[\mathcal{C}]$ such that, for every two of them, their neighborhood is Ω (see also Figure 4). Now for each $i \in \{1, 2, 3\}$, let $c_i = |C_i|$, $\overline{N}_i = \Omega \setminus N(C_i)$, $\bar{n}_i = |\overline{N}_i|$, $A = N(C_1) \cap N(C_2) \cap N(C_3)$, and $a = |A|$. Observe that $\overline{N}_1, \overline{N}_2, \overline{N}_3$, and A form a partition of Ω. Moreover, for any bijective mapping $\{i, j, k\}$ on $\{1, 2, 3\}$, we have that $N[C_i] = C_i \cup \overline{N}_j \cup \overline{N}_k \cup A$, and $N[C_j] \setminus N(C_i) = C_j \cup \overline{N}_i$. Therefore the sizes of the corresponding sets are as indicated in Part (5) of Algorithm 2. Hence the algorithm will enumerate two of the sets C_1, C_2, C_3 and generate the potential maximal clique Ω. ∎

It remains to prove that the running time of the algorithm is $\mathcal{O}(1.7347^n)$. We argue about each of the five cases of the algorithm separately.

(1) Minimal x, y-separators in $G_x y$ can be listed in time $\mathcal{O}(1.6181^n)$; see [37, Theorem 1]. Thus, cliques of the first type are enumerated in time $\mathcal{O}(1.6181^n)$.
(2) This part runs in $\mathcal{O}^*(\binom{n}{\alpha n})$ time. For $\alpha = 0.239$, this is bounded by $\mathcal{O}(1.7332^n)$.
(3) By Proposition 4.1, enumerating all connected vertex sets Z with $|N[Z]| \leq \beta n$ takes $\mathcal{O}^*(2^{\beta n})$ time. By putting $\beta = 0.794$, enumeration of potential maximal cliques of the third type takes time $\mathcal{O}(1.7339^n)$.
(4) With the previously fixed values α and β we have $\gamma < 8$ (see Claim 4), so $G[Z]$ will have at most 7 components. In order to produce all candidate sets Z we proceed as follows. First we use $O(n^7)$ time to guess one vertex of each component, that is we generate all vertex subsets Q of size at most 7. For each such set Q, let G^+ be obtained from G by adding edges between all vertices of Q. In this graph G^+, we enumerate all connected sets Z containing Q and such that $|Z| \leq (3/5)(n - |N(Z)|)$; this is done using Proposition 4.1 over all possible values of $|Z|$ and $N(Z)$. The number of such vertex sets Z can be bounded by $\mathcal{O}(1.7017^n)$. It only remains to test for each computed set Z that $N(Z)$ is a potential maximal clique.
(5) Generating all values c_1, c_2, c_3, $\bar{n}_1, \bar{n}_2, \bar{n}_3$, and a takes polynomial time. Note that for each tuple of values, all sets C_i are generated in time $\mathcal{O}^*(\binom{c_i + a + \bar{n}_j + \bar{n}_k}{c_i})$ using Proposition 4.1 (the binomial coefficient corresponds to $\binom{|N[C_i]|}{|C_i|}$; see also Figure 4). For each C_i we remove $N[C_i]$ from graph G and then generate all possible sets C_j, using again Proposition 4.1, in time $\mathcal{O}^*(\binom{c_j + \bar{n}_i}{c_j})$; here the binomial coefficient corresponds to $\binom{|N[C_j] \setminus N(C_i)|}{|C_j|}$. We point out that the previous best upper bound given in [37] was based on a somehow similar construction, but "guessing" the set $C_i \cup C_j$ in time $\binom{|N[C_i \cup C_j]|}{|C_i \cup C_j|}$. Our construction is faster since we always have that $\binom{|N[C_i]|}{|C_i|}\binom{|N[C_j] \setminus N(C_i)|}{|C_j|} \leq \binom{|N[C_i \cup C_j]|}{|C_i \cup C_j|}$.
It remains to argue that the maximum value of $\binom{c_i + a + \bar{n}_j + \bar{n}_k}{c_i}\binom{c_j + \bar{n}_i}{c_j}$, over all possible tuples, is at most 1.7347^n. This upper bound has been obtained using Stirling approximations and a computer program maximizing the function. The worst ratio occurs when $a = 0$, and for all $i \in \{1, 2, 3\}$ we have $c_i = 0.1768678n$ and $\bar{n}_i = (n - 3c_i)/3$. In this case we obtain a maximum value of type η^n, for $\eta < 1.7347$.

This concludes the proof of Theorem 1.1.

5. Applications. In this section we discuss several applications of Theorem 3.1. We start by mentioning the most interesting special cases of the optimization problem (1.1). 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 (1.1) is trivial but for some it is nonobvious and requires deep results from graph theory. Our results are summarized in Theorem 5.1.

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 $\mathcal{F}$ containing a planar graph, 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 $\mathcal{F}$.

The assumption that $\mathcal{F}$ contains a planar graph is crucial, because it ensures (see Proposition 5.7) that the solution subgraph $G[F]$ is of bounded treewidth.

Even the special case with $\ell = 0$, this problem and its complementary version called the Minimum $\mathcal{F}$-Deletion, encompasses many different problems. In the literature, the case $\ell = 0$ was studied from parameterized and approximation perspectives [34].

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 the 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 pseudoforest 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 the minimum number of vertices in order to obtain a graph of treewidth at most η. Since by the result of Robertson and Seymour [62] 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 outer-planar 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 [17]. Recall that Cameron and Hell defined a 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 are 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 an annotated version of optimization problem (1.1). Note that k is not required to be a constant.

k-in-a-Graph From $\mathcal{G}(t, \varphi)$
Input: A graph G, with k annotated vertices.
Task: Find an induced graph from $\mathcal{G}(t, \varphi)$ containing all k annotated vertices.

It is also easy to handle variants of this problem where the annotated 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 graphs of maximum vertex degree at most three [28]. Bienstock [6] has 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 [20]. The complexity of the case $k = 4$ is open.

Theorem 5.1. *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 Theorem 1.1, Theorem 5.1 implies the following.

Corollary 5.2. *Let* G *be an* n*-vertex graph. All problems from Theorem* 5.1 *are solvable in time* $\mathcal{O}(1.7347^n)$.

Theorem 5.1 follows from Theorem 3.1 and Lemmas 5.4, 5.6, 5.8, 5.9, and 5.10.

Let us remark that Theorem 5.1 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 D, 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 (mod m). We will need the following result of Thomassen.

Proposition 5.3 (see [69]). *For all 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 5.3, we obtain the following lemma.

Lemma 5.4. 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 5.3. By Proposition 5.3, 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, Bondy, and Reed.

Proposition 5.5 (see [7]). *Graphs without* ℓ *disjoint cycles of length at least* p *are of treewidth* $\mathcal{O}(\ell^2 p)$.

By making use of Proposition 5.5, it is easy to prove the following lemma.

Lemma 5.6. 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 5.5, 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 the 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 [63]; see also [64].

PROPOSITION 5.7 (see [63]). *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 5.8. *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 5.7, 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 5.9. 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 fact that a vertex set C belongs to $cc(G[F])$ can be expressed by an MSO formula (see, e.g., [11]), we have that the following property is regular:

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

***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 3.9). The following lemma follows from the definition of the problems.

LEMMA 5.10. k-IN-A-GRAPH FROM $\mathcal{G}(t, \varphi)$ *is a special case of* OPTIMAL WEIGHTED ANNOTATED INDUCED SUBGRAPH FOR $\mathcal{P}$ AND t

6. Graph classes. In this section we discuss the consequences of Theorem 5.1 for special graph classes. In particular, by Proposition 2.8, 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., a 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 [51]. The class of d-trapezoid graphs is defined as follows. Let $L_1, \ldots, L_d$

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 [53]; see also [15]. An intersection graph of polygons enclosed by a bounding circle is is know as a *polygon-circle graph*. As was observed by Suchan in [67], every polygon circle with n vertices has $\mathcal{O}(n^2)$ minimal separators. See Figure 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 Figure 1.

Let us remark that the only information we need for our algorithms 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 types of algorithms were called *robust* by Raghavan and Spinrad [60]. For example, while recognition of d-trapezoid and polygon-circle graphs is NP-complete [72, 59], our algorithm either correctly solves the problem or outputs that the input graph is not d-trapezoid or polygon-circle.

COROLLARY 6.1. *All problems from Theorem* 5.1 *are solvable in polynomial time on classes of graphs from Figure* 1.

On several classes of graphs even more general problems can be solved. The observation here is that for many classes of graphs from Figure 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 variants of colorings, we need the the following proposition due to Gaspers et al.

PROPOSITION 6.2 (see [40]). *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 6.2, Theorem 5.1 allows us to show that on several graph classes, in addition to problems encompassed by Corollary 6.1, an 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 6.3. *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 [30], 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 the $(k \times k)$-grid 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 [51] 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 6.3 with Theorem 5.1, 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.

7. Conclusion.

In this paper we have shown how the theory of minimal triangulations combined with the power of CMSO logic can be used to obtain moderate exponential and polynomial algorithms for various problems about induced subgraphs.

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 annotated 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 annotated vertices x_i, y_i, $1 \leq i \leq p$, the task is to find a set of paths connecting each x_i to each y_i 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 an already very special case of the 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 $n^{\mathcal{O}(t)}$.

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

Similarly, when it concerns exact exponential algorithms, in many cases the requirements of Theorem 3.9 that the treewidth t of the induced subgraph is a constant, can be relaxed. For example, if we want to find a maximum induced subgraph of treewidth at most t, the running time of the dynamic programming algorithm used to prove this theorem would be $\mathcal{O}(1.7347^n n^t)$, which is $\mathcal{O}(1.7347^n)$ for $t = o(n/\log n)$. This observation, combined with specific properties of planar potential maximal cliques and careful implementation of dynamic programming, was used in [36] to obtain an $\mathcal{O}(1.7347^n)$-time algorithm computing a maximum induced planar subgraph. Interestingly enough, no algorithm breaking the brute force $\mathcal{O}(2^n)$-barrier for finding a maximum induced subgraph excluding some fixed graph H as a minor is known in the literature.

Since the appearance of the preliminary version of this paper, several new applications of potential maximal cliques were discovered. For polynomial-time algorithms, a nontrivial extension of our approach was used by Lokshtanov, Vatshelle, and Villanger [55] to settle the long standing open problem about the complexity of MAXIMUM INDEPENDENT SET on P_5-free graphs. In parameterized algorithms, potential maximal cliques were used to obtain a subexponential parameterized algorithm for the MINIMUM FILL-IN problem [38].

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.

A 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, our approach works without a problem for Maximum Induced Tree, where the task is to find a minimum feedback vertex set such that the remaining graph is connected, i.e., a tree.

Another interesting question is how many potential maximal cliques can be in an n-vertex graph? By Theorem 1.1, we know that it is at most $\mathcal{O}(1.7347^n)$. The worst case in the proof of Theorem 1.1 occurs in the enumeration of potential maximal cliques of the fifth type, and this case can be improved as follows. Let us remind ourselves that in this case, we have three connected components C_1, C_2, C_3 of $G - \Omega$ equal sizes. Also the sizes or neighborhoods $N(C_i)$ are, for each $i \in \{1, 2, 3\}$, equal to $(2/3)|\Omega|$. In this situation, the set $N(C_1) \cap N(C_2)$ is "invisible" from C_3, and thus is internal to the graph $N[C_1 \cup C_2]$. In this situation, by guessing a vertex of $u \in C_1$ and $v \in C_2$, we can use the firefighters lemma in the graph $G + uv$ obtained from G by connecting u and v in order to enumerate all connected sets of size $|C_1| + |C_2| + |C_1 \cap C_2|$ with neighborhoods of size $|C_3|/2$. By doing this, one can compute component C_3 in time better than $\mathcal{O}(1.7347^n)$. Similarly, we can compute C_1, and thus Ω. Unfortunately, these arguments fall apart if Ω contains vertices which can "see" three components. To make them work, we have to consider more detailed balancing, by first guessing the vertices seeing the three components. By implementing this idea accurately, and by the cost of more technical arguments, it is possible to improve the running time of the algorithm slightly (in the third digit after the dot in the base of the exponent). But how close is the bound $\mathcal{O}(1.7347^n)$ to the truth? There are graphs with roughly $3^{n/3} \approx 1.442^n$ potential maximal cliques [33]. Let us remind ourselves that by the classical result of Moon and Moser [57] (see also Miller and Muller [56]), the number of maximal cliques in a graph on n vertices is at most $3^{n/3}$. Can it be that the right upper bound on the number of potential maximal cliques is also of order $3^{n/3}$? Can we enumerate potential maximal cliques within this time? By Theorem 3.9, this would yield a dramatic improvement for numerous exact algorithms.

Acknowledgments. 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 Appl. Math., 44 (1993), pp. 185–190.

[2] S. R. Arikati, C. Pandu 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*, J. 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), Lecture Notes in Comput. Sci. 7074, Springer, Berlin, 2011, pp. 110–119.

[5] A. Berry, J. P. Bordat, and O. Cogis, *Generating all the minimal separators of a graph*, Internat. 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 Math., 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*, Theoret. 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, IEEE, Los Alamitos, CA, 2009, 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] V. Bouchitté and I. Todinca, *Listing all potential maximal cliques of a graph*, Theoret. Comput. Sci., 276 (2002), pp. 17–32.
[14] N. Bourgeois, B. Escoffier, V. Th. Paschos, and J. M. M. van Rooij, *Fast algorithms for max independent set*, Algorithmica, 62 (2012), pp. 382–415.
[15] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes. A Survey*, SIAM Monogr. Discrete Math. Appl., SIAM, Philadelphia, 1999.
[16] K. Cameron, *Induced matchings*, Discrete Appl. Math., 24 (1989), pp. 97–102.
[17] K. Cameron and P. Hell, *Independent packings in structured graphs*, Math. Program., 105 (2006), pp. 201–213.
[18] K. Cameron, R. Sritharan, and Y. Tang, *Finding a maximum induced matching in weakly chordal graphs*, Discrete Math., 266 (2003), pp. 133–142.
[19] J.-M. Chang, *Induced matchings in asteroidal triple-free graphs*, Discrete Appl. Math., 132 (2003), pp. 67–78.
[20] M. Chudnovsky and P. D. Seymour, *The three-in-a-tree problem*, Combinatorica, 30 (2010), pp. 387–417.
[21] B. Courcelle, *The monadic second-order logic of graphs* I: *Recognizable sets of finite graphs*, Inform. and Comput., 85 (1990), pp. 12–75.
[22] B. Courcelle, *The monadic second-order logic of graphs.* III. *Tree-decompositions, minors and complexity issues*, RAIRO Inform. Théoret. Appl., 26 (1992), pp. 257–286.
[23] B. Courcelle, *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 Science, River Edge, NJ, 1997, pp. 313–400.
[24] B. Courcelle and J. Engelfriet, *Graph Structure and Monadic Second-Order Logic*, Cambridge University Press, Cambridge, 2012.
[25] B. Courcelle and M. Mosbah, *Monadic second-order evaluations on tree-decomposable graphs*, Theoret. Comput. Sci., 109 (1993), pp. 49–82.
[26] M. Davis and H. Putnam, *A computing procedure for quantification theory*, J. ACM, 7 (1960), pp. 201–215.
[27] 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), IEEE, Los Alamitos, CA, 2007, pp. 270–279.
[28] N. Derhy and C. Picouleau, *Finding induced trees*, Discrete Appl. Math., 157 (2009), pp. 3552–3557.
[29] 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.
[30] F. V. Fomin, P. A. Golovach, and D. M. Thilikos, *Contraction obstructions for treewidth*, J. Combin. Theory Ser. B, 101 (2011), pp. 302–314.
[31] F. V. Fomin, F. Grandoni, and D. Kratsch, *A measure & conquer approach for the analysis of exact algorithms*, J. ACM, 56 (2009), 25.
[32] F. V. Fomin and D. Kratsch, *Exact Exponential Algorithms*, Springer, Heidelberg, 2010.
[33] 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.
[34] 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, Piscataway, NJ, 2012, pp. 470–479.
[35] 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, Philadelphia, 2010, pp. 503–510.

[36] F. V. Fomin, I. Todinca, and Y. Villanger, *Exact algorithm for the maximum induced planar subgraph problem*, in Proceedings of 19th Annual European Symposium on Algorithms (ESA), Lecture Notes in Comput. Sci. 6942, Springer, Berlin, 2011, pp. 287–298.

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

[38] F. V. Fomin and Y. Villanger, *Subexponential parameterized algorithm for minimum fill-in*, SIAM J. Comput., 42 (2013), pp. 2197–2216.

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

[40] S. Gaspers, D. Kratsch, M. Liedloff, and I. Todinca, *Exponential time algorithms for the minimum dominating set problem on some graph classes*, ACM Trans. Algorithms, 6 (2009), 9.

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

[42] F. Gavril, *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*, Theoret. 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 Appl. Math., 101 (2000), pp. 157–165.

[46] M. Grohe, *Logic, graphs, and algorithms*, in Logic and Automata-History and Perspectives, J. Flum, E. Grädel, and T. Wilke, eds, 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 Math., 306 (2006), pp. 297–317.

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

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

[52] T. Kloks, C.-H. Liu, and S.-H. Poon, *Feedback Vertex Set on Chordal Bipartite Graphs*, preprint, arXiv:1104.3915, 2012.

[53] D. Kratsch, *The Structure of Graphs and the Design of Efficient Algorithms*, Habilitation thesis, F. Schiller Universität, Jena, Germnay, 1996.

[54] S. Kreutzer, *Algorithmic meta-theorems*, in Finite and Algorithmic Model Theory, London Math. Soc. Lecture Note Ser. 379, Cambridge University Press, Cambridge, 2011, pp. 177–270.

[55] D. Lokshtanov, M. Vatshelle, and Y. Villanger, *Independent set in P_5-free graphs in polynomial time*, in Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithm (SODA), SIAM, Philadelphia, 2014, pp. 570–581.

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

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

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

[59] 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), Lecture Notes in Comput. Sci. 4769, Springer, Berlin, 2007, pp. 238–247.

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

[61] I. Razgon, *Exact computation of maximum induced forest*, in Proceedings of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006), Lecture Notes in Comput. Sci. 4059, Springer, Berlin, 2006, pp. 160–171.

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

[63] N. Robertson and P. D. Seymour, *Graph minors. V. Excluding a planar graph*, J. Combin. Theory Ser. B, 41 (1986), pp. 92–114.

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

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

[66] 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.

[67] K. Suchan, *Minimal Separators in Intersection Graphs*, Master's thesis, Akademia Gorniczo-Hutnicza im. Stanislawa Staszica w Krakowie, Cracow, 2003.

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

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

[70] M. Xiao and H. Nagamochi, *Exact algorithms for maximum independent set*, in Algorithms and Computation 24th International Symposium, ISAAC 2013, Hong Kong, China, 2013, L. Cai, S.-W. Cheng, and T. W. Lam, eds., Lecture Notes in Comput. Sci. 8283, Springer, Berlin, 2013, pp. 328–338.

[71] M. Xiao and H. Nagamochi, *An improved exact algorithm for undirected feedback vertex set*, in In proceedings of the 7th Combinatorial Optimization and Applications (COCOA), Lecture Notes in Comput. Sci. 8287, Springer, Berlin, 2013, pp. 153–164.

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

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