Algorithmica (2018) 80:1146–1169
https://doi.org/10.1007/s00453-017-0297-1

Algorithms Parameterized by Vertex Cover and Modular Width, Through Potential Maximal Cliques

Fedor V. Fomin[1] · **Mathieu Liedloff**[2] · **Pedro Montealegre**[2] · **Ioan Todinca**[2]

Received: 26 May 2016 / Accepted: 20 February 2017 / Published online: 27 February 2017

Abstract In this paper we give upper bounds on the number of *minimal separators* and *potential maximal cliques of graphs* w.r.t. two graph parameters, namely *vertex cover* (vc) and *modular width* (mw). We prove that for any graph, the number of its minimal separators is $\mathcal{O}^*(3^{\mathrm{vc}})$ and $\mathcal{O}^*(1.6181^{\mathrm{mw}})$, and the number of potential maximal cliques is $\mathcal{O}^*(4^{\mathrm{vc}})$ and $\mathcal{O}^*(1.7347^{\mathrm{mw}})$, and these objects can be listed within the same running times (The $\mathcal{O}^*$ notation suppresses polynomial factors in the size of the input). Combined with known applications of potential maximal cliques, we deduce that a large family of problems, e.g., Treewidth, Minimum Fill- in, Longest Induced Path, Feedback vertex set and many others, can be solved in time $\mathcal{O}^*(4^{\mathrm{vc}})$ or $\mathcal{O}^*(1.7347^{\mathrm{mw}})$. With slightly different techniques, we prove that the Treedepth problem can be also solved in single-exponential time, for both parameters.

Keywords Parametrized algorithms · Potential maximal cliques · Treewidth · Vertex cover

Part of these results appeared in a preliminary version of this paper [17], in the proceedings of SWAT 2014. Partially supported by the ANR project GraphEn, ANR-15-CE40-0009.

✉ Ioan Todinca
ioan.todinca@univ-orleans.fr

Fedor V. Fomin
fedor.fomin@ii.uib.no

Mathieu Liedloff
mathieu.liedloff@univ-orleans.fr

Pedro Montealegre
pedro.montealegre@univ-orleans.fr

1 Department of Informatics, University of Bergen, 5020 Bergen, Norway

2 Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, BP 6759, 45067 Orléans Cedex 2, France

1 Introduction

The *vertex cover* of a graph G, denoted by $\mathrm{vc}(G)$, is the minimum number of vertices that cover all edges of the graph. The *modular width* $\mathrm{mw}(G)$ of graph $G = (V, E)$ is related to modular decompositions (see Sect. 4 and [30] for definitions). A *module* of G is a set of vertices $M \subseteq V$ such that, for any vertex x outside the module, x is either adjacent to all vertices of M, or to none of them. A *prime graph* is a graph that has only trivial modules (i.e., formed by a single vertex or the whole vertex set). The modular width of G is the size of a largest prime induced subgraph of G.

Minimal separators and *potential maximal cliques* are strongly related to *minimal triangulations* of graphs. A *triangulation* of an arbitrary graph $G = (V, E)$ is a chordal supergraph $H = (V, E')$, i.e., a supergraph on the same vertex set, having no induced cycles with strictly more than 3 vertices. We say that H is a *minimal triangulation* is G has no triangulation strictly contained in H, as a subgraph. A set of vertices Ω of G is called a *potential maximal clique* if Ω induces a maximal clique in some minimal triangulation H of G.

The main results of this paper are of combinatorial nature: we show that the number of minimal separators and the number of potential maximal cliques of a graph are upper bounded by a single-exponential function in each of the parameters vertex cover and modular width. More specifically, we prove the number of minimal separators in a graph G is at most $3^{\mathrm{vc}(G)}$ and $\mathcal{O}^*(1.6181^{\mathrm{mw}(G)})$, and the number of potential maximal cliques is $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ and $\mathcal{O}^*(1.7347^{\mathrm{mw}(G)})$. Moreover, these objects can be listed within the same running time bounds. Recall that the $\mathcal{O}^*$ notation suppresses polynomial factors in the size of the input, i.e., $\mathcal{O}^*(f(k))$ should be read as $f(k) \cdot n^{\mathcal{O}(1)}$ where n is the number of vertices of the input graph. Minimal separators and potential maximal cliques have been used for solving several classical optimization problems, e.g., Treewidth, Minimum Fill-In [16], Longest Induced Path, Feedback Vertex Set or Independent Cycle Packing [18]. Pipelined with our combinatorial bounds, we obtain a series of algorithmic consequences in the area of FPT algorithms parameterized by the vertex cover and the modular width of the input graph. In particular, the problems mentioned above can be solved in time $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ and $\mathcal{O}^*(1.7347^{\mathrm{mw}(G)})$. These results are complementary in the sense that graphs with small vertex cover are sparse, while graphs with small modular width may be dense.

Vertex cover and modular width are strongly related to treewidth (tw) and cliquewidth (cw) parameters. It is an easy exercice to show that, since for any graph G, we have $\mathrm{tw}(G) \le \mathrm{vc}(G)$ and $\mathrm{cw}(G) \le \mathrm{mw}(G) + 2$; see [22] for more parameters and relations between them. The celebrated theorem of Courcelle [9] states that all problems expressible in Counting Monadic Second Order Logic (CMSO_2) can be solved in time $f(\mathrm{tw}) \cdot n$ for some function f depending on the problem. A similar result for cliquewidth [11] shows that all CMSO_1 problems can be solved in time $f(\mathrm{cw}) \cdot n$, if the clique-decomposition is also given as part of the input (See the “Appendix 2’ for definitions of different types of logic. Informally, CMSO_2 allows logic formulae with quantifiers over vertices, edges, edge sets and vertex sets, and counting modulo constants. The CMSO_1 formulae are more restricted, we are not allowed to quantify over edge sets).

Typically function f is a tower of exponentials, and the height of the tower depends on the formula. Moreover Frick and Grohe [21] proved that this dependency on the treewidth/cliquewidth of the input graph G cannot be significantly improved in general. Lampis [24] shows that the running time for CMSO_2 problems can be improved $2^{2^{\mathcal{O}(\text{vc}(G))}} \cdot n$ when parametrized by vertex cover, but he also shows that this cannot be improved to $\mathcal{O}^*(2^{2^{o(\text{vc}(G))}})$ (under the exponential time hypothesis). We are not aware of similar improvements for the modular width parameter, but we refer to [22] for discussions on problems parameterized by modular width.

Most of our algorithmic applications concern a restricted, though still large subset of CMSO_2 problems, but we guarantee algorithms that are single exponential in the vertex cover: $\mathcal{O}^*(4^{\text{vc}(G)})$ and in the modular width: $\mathcal{O}^*(1.7347^{\text{mw}(G)})$. We point out that our result for modular width extends the result of [18,19], who show a similar bound of $\mathcal{O}^*(1.7347^n)$ for the number of potential maximal cliques and for the running times for these problems, but parameterized by the number of vertices of the input graph.

Maximum induced graph of bounded treewidth We use the following generic problem proposed by [18], that encompasses many classical optimization problems. Fix an integer $t \geq 0$ and a CMSO_2 formula φ. Consider the problem of finding, in the input graph G, an induced subgraph $G[F]$ together with a vertex subset $X \subseteq F$, such that the treewidth of $G[F]$ is at most t, the graph $G[F]$ together with the vertex subset X satisfying formula φ, and X is of maximum size under these conditions. This optimization problem is called Max Induced Subgraph of tw $\leq t$ satisfiying φ:

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

Note that our formula φ has a free variable corresponding to the vertex subset X. For several examples, in formula φ the vertex set X is actually equal to F. E.g., even when φ only states that $X = F$, for $t = 0$ we obtain the Maximum Independent set problem, and for $t = 1$ we obtain the Maximum Induced Forest (and in this case $V \setminus F$ is an optimal solution for Feedback Vertex Set). If $t = 1$ and φ states that $X = F$ and $G[F]$ is a path we obtain the Longest Induced Path problem. Still under the assumption that $X = F$, we can express the problem of finding the largest induced subgraph $G[F]$ excluding a fixed planar graph H as a minor, or the largest induced subgraph with no cycles of length $0 \mod l$. But X can correspond to other parameters, e.g. we can choose the formula φ such that $|X|$ is the number of connected components of $G[F]$. Based on this we can express problems like Independent Cycle Packing, where the goal is to find an induced subgraph with a maximum number of components, and such that each component induces a cycle.

By the result of [18], Max Induced Subgraph of tw $\leq t$ satisfiying φ is solvable in time $\#\,\text{pmc} \cdot n^{t+4} \cdot f(\varphi, t)$ where $\#\,\text{pmc}$ is the number of potential maximal cliques of the graph, assuming that the set of all potential maximal cliques is also part of the input, and f is some function of φ and t only. Thanks to our combinatorial bounds, the problem MAX INDUCED SUBGRAPH OF tw $\leq t$ SATISFIYING φ can be solved in times $\mathcal{O}(4^{\mathrm{vc}(G)} n^{t+c})$ and $\mathcal{O}(1.7347^{\mathrm{mw}(G)} n^{t+c})$, for some small constant c.

Treewidth, treedepth and other parameters Several other graph problems can be solved in time $\mathcal{O}^*(\#\,\mathrm{pmc})$ if the input graph is given together with the set of its potential maximal cliques. Examples of such problems are TREEWIDTH, MINIMUM FILL- IN [16], their weighted versions [3,23] and several problems related to phylogeny [23], or TREELENGTH [26]. Pipelined with our main combinatorial result, we deduce that all these problems can be solved in time $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ or $\mathcal{O}^*(1.7347^{\mathrm{mw}(G)})$. Chapelle et al. [7] provided an algorithm solving TREEWIDTH and PATHWIDTH in $\mathcal{O}^*(3^{\mathrm{vc}(G)})$, but those completely different techniques do not seem to work for MINIMUM FILL- IN or TREELENGTH. The interested reader may also refer., e.g., to [12,14] for more (layout) problems parameterized by vertex cover. The best known parameterized algorithm deciding whether a given n-vertex graph G is of treewidth at most t runs in time $\mathcal{O}(t^{t^3} n)$ [2]. The treewidth of a graph can be computed in time $\mathcal{O}(1.7347^n)$ [18].

Treedepth is a graph parameter that encountered a regain of interest in the area of sparse graphs, cf. the book of Nešetřil and Ossona de Mendes [27]. Actually the parameter used to be known under different names, e.g., *vertex ranking* [13]. A graph has treedepth (vertex ranking) at most t if its vertices can be labeled from 1 to t, such that each path connecting vertices of the same label i contains an internal vertex with a label greater than i. The parameter is NP-hard to compute, but one can decide if the treedepth of an n-vertex graph is at most t in time $2^{\mathcal{O}(t^2)} \cdot n$ [28]. The treedepth can be also computed in time $\mathcal{O}(1.9602^n)$ [15].

Deogun et al. [13] provide polynomial algorithms computing the vertex ranking for several graph classes, using minimal separators. Based on their approach and new combinatorial bounds we show that the treedepth of a graph can also be computed in parameterized single-exponential time, when parameterized by the vertex cover or by the modular width of the input graph.

The paper is organized as follows. Section 2 introduces the preliminary results on minimal triangulations, minimal separators and potential maximal cliques. Section 3 presents the combinatorial upper bounds on the number of minimal separators and potential maximal cliques, with respect to vertex cover. Section 4 provides similar bounds, with respect to modular width. Applications of these results are presented in Sect. 5. The results for treedepth do not rely directly on potential maximal cliques but on a different tool; they can be found in Sect. 6. Conclusion section discusses further research directions.

2 Minimal Separators and potential maximal cliqueS

Let $G = (V, E)$ be an undirected, simple graph. We denote by n its number of vertices and by m its number of edges. The *neighborhood* of a vertex v is $N(v) = \{u \in V : \{u, v\} \in E\}$. We say that a vertex x *sees* a vertex subset S (or vice-versa) if $N(x)$ intersects S. For a vertex set $S \subseteq V$ we denote by $N(S)$ the set $\bigcup_{v \in S} N(v) \setminus S$.

We write $N[S]$ (resp. $N[x]$) for $N(S) \cup S$ (resp. $N(x) \cup \{x\}$). Also $G[S]$ denotes the subgraph of G induced by S, and $G - S$ is the graph $G[V \setminus S]$.

A *connected component* of graph G is the vertex set of a maximal induced connected subgraph of G. Consider a vertex subset S of graph G. Given two vertices u and v, we say that S is a u, v-separator if u and v are in different connected components of $G - S$. Moreover, if S is inclusion-minimal among all u, v-separators, we say that S is a *minimal* u, v*-separator*. A vertex subset S is called a *minimal separator* of G if S is a u, v-minimal separator for some pair of vertices u and v.

Let C be a component of $G - S$. If $N(C) = S$, we say that C is a *full component* associated to S.

Proposition 1 (folklore) *A vertex subset S of G is a minimal separator if $G - S$ has at least two full components associated to S. Moreover, S is a minimal x, y-separator if and only if x and y are in different full components associated to S.*

A graph H is *chordal* or *triangulated* if every cycle with four or more vertices has a chord, i.e., an edge between two non-consecutive vertices of the cycle. 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.

A set of vertices $\Omega \subseteq V$ 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.

The following statement due to Bouchitté and Todinca [5] provides a characterization of potential maximal cliques, and in particular allows to test in polynomial time if a vertex subset Ω is a potential maximal clique of G:

Proposition 2 ([5]) *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 denote $\mathcal{S}(\Omega) = \{S_1, S_2, \dots, S_p\}$, where $S_i = N(C_i)$ for all $i \in \{1, \dots, p\}$. 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.

Another way of stating the second condition is that for any pair of vertices $u, v \in \Omega$, if they are not adjacent in G then there is a component C of $G - \Omega$ seeing both x and y.

To illustrate Proposition 2, consider, e.g., the cube graph depicted in Fig. 1. The set $\Omega_1 = \{a, e, g, c, h\}$ is a potential maximal clique and the minimal separators contained in Ω_1 are $\{a, e, g, c\}$ and $\{a, h, c\}$. Another potential maximal clique of the cube graph is $\Omega_2 = \{a, c, f, h\}$ containing the minimal separators $\{a, c, f\}$, $\{a, c, h\}$, $\{a, f, h\}$ and $\{c, f, h\}$.

Based on Propositions 1 and 2, one can easily deduce:

Corollary 1 (see e.g., [5]) *There is an $O(m)$ time algorithm testing if a given vertex subset S is a minimal separator of G, and $O(nm)$ time algorithm testing if a given vertex subset Ω is a potential maximal clique of G.*

We also need the following observation.

Proposition 3 ([5]) *Let Ω be a potential maximal clique of G and let $S \subset \Omega$ be a minimal separator. Then $\Omega \setminus S$ is contained into a unique component C of $G - S$, and moreover C is a full component associated to S.*

3 Relations to Vertex Cover

A vertex subset W is a *vertex cover* of G if each edge has at least one endpoint in W. Note that if W is a vertex cover, then $V \setminus W$ induces an *independent set* in G, i.e., $G - W$ contains no edges. We denote by $\mathrm{vc}(G)$ the size of a minimum vertex cover of G. The parameter $\mathrm{vc}(G)$ is called the *vertex cover number* or simply (by a slight abuse of language) the *vertex cover* of G. After a long sequence of improvements, the current fastest parameterized algorithm for Vertex Cover is the algorithm of Chen, Kanj, and Xia, running in time $\mathcal{O}(1.2738^{\mathrm{vc}(G)} + n\,\mathrm{vc}(G))$ [8]. However, for our purposes even a weaker result will do the job.

Proposition 4 (folklore) *There is an algorithm computing the vertex cover of the input graph G in time $\mathcal{O}^*(2^{\mathrm{vc}(G)})$.*

Let us show that any graph G has at most $3^{\mathrm{vc}(G)}$ minimal separators.

Lemma 1 *Let $G = (V, E)$ be a graph, W be a vertex cover and $S \subseteq V$ be a minimal separator of G. Consider a three-partition (D_1, S, D_2) of V such that both D_1 and D_2 are formed by a union of components of $G - S$, and both D_1 and D_2 contain some full component associated to S. Denote $D_1^W = D_1 \cap W$ and $D_2^W = D_2 \cap W$.*

Then $S \setminus W = \{x \in V \setminus W \mid N(x)$ intersects both D_1^W and $D_2^W\}$.

Proof Let $C_1 \subseteq D_1$ and $C_2 \subseteq D_2$ be two full components associated to S. Let $x \in S \setminus W$. Vertex x must have neighbors both in C_1 and C_2, hence both in D_1 and D_2. Since $x \notin W$ and W is a vertex cover, we have $N(x) \subseteq W$. Consequently x has neighbors both in D_1^W and D_2^W.

Conversely, let $x \in V \setminus W$ s.t. $N(x)$ intersects both D_1^W and D_2^W. We prove that $x \in S$. By contradiction, assume that $x \notin S$, thus x is in some component C of $G - S$. Suppose w.l.o.g. that $C \subseteq D_1$. Since $N(x) \subseteq C \cup N(C)$, we must have $N(x) \subseteq D_1 \cup S$. Thus $N(x)$ cannot intersect D_2—a contradiction. □

Theorem 1 *Any graph G has at most $3^{\mathrm{vc}(G)}$ minimal separators. Moreover the set of its minimal separators can be listed in $\mathcal{O}^*(3^{\mathrm{vc}(G)})$ time.*

Proof Let W be a minimum size vertex cover of G. For each three-partition (D_1^W, S^W, D_2^W) of W, let $S = S^W \cup \{x \in V \setminus W \mid N(x)$ intersects D_1^W and $D_2^W\}$. According to Lemma 1, each minimal separator of G will be generated this way, by an

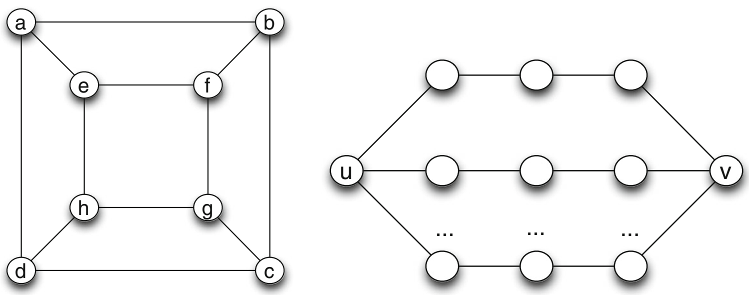

Fig. 1 Cube graph (*left*) and watermelon graph (*right*)

appropriate partition (D_1^W, S^W, D_2^W) of W. Thus the number of minimal separators is at most $3^{\mathrm{vc}(G)}$, the number of three-partitions of W.

These arguments can be easily turned into an enumeration algorithm, we simply need to compute an optimum vertex cover then test, for each set S generated from a three-partition, if S is indeed a minimal separator. The former part takes $\mathcal{O}^*(2^{\mathrm{vc}(G)})$ time by Proposition 4, and the latter takes polynomial time for each set S using Corollary 1. □

Observe that the bound of Theorem 1 is tight up to a constant factor. Indeed consider the watermelon graph $W_{k,3}$ formed by k disjoint paths of three vertices plus two vertices u and v adjacent to the left, respectively right ends of the paths (see Fig. 1). Note that this graph has vertex cover $k + 2$ (the minimum vertex cover contains the middle of each path and vertices u and v) and it also has 3^k minimal u, v-separators, obtained by choosing arbitrarily one of the three vertices on each of the k paths.

We now extend Theorem 1 to a similar result on potential maximal cliques. Let us distinguish a particular family of potential maximal cliques, which have *active* separators. They have a particular structure which makes them easier to handle.

Definition 1 ([6]) Let $\Omega \subseteq V$ be a potential maximal clique of graph $G = (V, E)$, let $\{C_1, \ldots, C_p\}$ be the set of connected components of $G - \Omega$ and let $S_i = N(C_i)$, for $1 \leq i \leq p$.

Consider the graph G^+ obtained from G by completing into a clique all minimal separators S_j, $2 \leq j \leq p$, such that S_j is not contained in S_1.

We say that S_1 is an *active separator* for Ω if Ω is not a clique in this graph G^+. A pair of vertices $x, y \in \Omega$ that are not adjacent in G^+ is called an *active pair*. Note that, by Proposition 2, we must have $x, y \in S_1$.

The following statement characterizes potential maximal cliques with active separators.

Proposition 5 *Let Ω be a potential maximal clique having an active separator $S \subset \Omega$, with an active pair $x, y \in S$. Denote by C the unique component of $G - S$ containing $\Omega \setminus S$. Then $\Omega \setminus S$ is a minimal x, y-separator in the graph $G[C \cup \{x, y\}]$.*

Again on the cube graph of Fig. 1, for the potential maximal clique $\Omega_1 = \{a, e, g, c, h\}$, both minimal separators are active. E.g., for the minimal separator

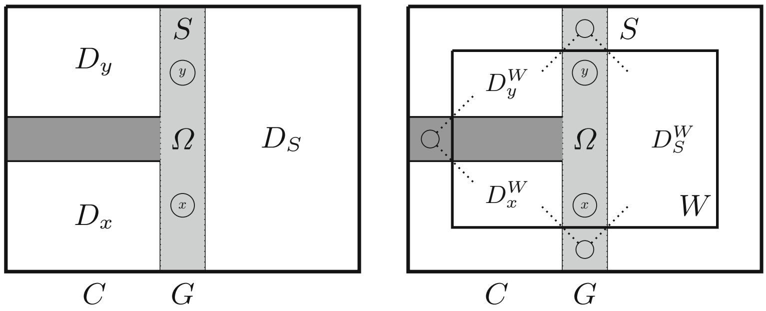

Fig. 2 Four-partition of V (*left*) and a four-partition of W (*right*)

$S = \{a, e, g, c\}$ the pair $\{e, g\}$ is active. Not all potential maximal cliques have active separators, as illustrated by the potential maximal clique $\Omega_2 = \{a, c, f, h\}$ of the same graph.

Let us first focus on potential maximal cliques having an active separator. We give a result similar to Lemma 1, showing that such a potential maximal clique can be determined by a certain partition of the vertex cover W of G.

Lemma 2 *Let $G = (V, E)$ be a graph and W be a vertex cover of G. Consider a potential maximal clique Ω of G having an active separator $S \subset \Omega$ and an active pair $x, y \in S$. Let C be the unique connected component of $G - S$ intersecting Ω and let D_S be the union of all other connected components of $G - S$. Denote by D_x the union of components of $G - \Omega$ contained in C, seeing x, by D_y the union of components of $G - \Omega$ contained in C not seeing x.*

Now let $D_S^W = D_S \cap W$, $D_x^W = D_x \cap W$ and $D_y^W = D_y \cap W$.

Then one of the following holds:

1. *There is a vertex $t \in \Omega$ such that $\Omega \setminus S = N(t) \cap C$.*
2. *There is a vertex $t \in \Omega$ such that $\Omega = N[t]$.*
3. *A vertex $z \notin W$ is in Ω if and only if*
   (a) *z sees D_S^W and $D_x^W \cup D_y^W$, or*
   (b) *z does not see D_S^W but it sees $D_x^W \cup \{x\}$, $D_y^W \cup \{y\}$ and $D_x^W \cup D_y^W$.*

Proof Note that D_x, D_y, D_S and Ω form a partition of the vertex set V. This induces a four-partition of the vertex cover W (see Fig. 2).

We first prove that any vertex $z \notin W$ satisfying conditions 3a or 3b must be in Ω.

Consider first the case 3a when z sees D_S^W and $D_x^W \cup D_y^W$. So z sees D_S and C; we can apply Lemma 1 to partition (D_S, S, C) thus $z \in S$. Consider now the case 3b when z sees $D_x^W \cup D_y^W$, $D_x \cup \{x\}$ and $D_y \cup \{y\}$ but not D_S^W. Again by Lemma 1 applied to partition (D_S, S, C), vertex z cannot be in S. Since z has a neighbor in $D_x \cup D_y$, we have $z \in C$. Let $H = G[C \cup \{x, y\}]$ and $T = \Omega \cap C$ (thus we also have $T = \Omega \setminus S$). Recall that T is an x, y-minimal separator in H by Proposition 5. By definition of set D_x, we have that $D_x \cup \{x\}$ is exactly the component of $H - T$ containing x. Note that $D_y \cup \{y\}$ is the union of the component of $H - T$ containing y and of all other components of $H - T$ (that no not see x nor y). By applying Lemma 1 on graph H, with vertex cover $(W \cap C) \cup \{x, y\}$ and with partition $(D_x \cup \{x\}, T, D_y \cup \{y\})$ we deduce that $z \in T$.

Conversely, let $z \in \Omega \setminus W$. We must prove that either z satisfies conditions 3a or 3b, or we are in one of the first two cases of the Lemma. We distinguish the cases $z \in S$ and $z \in T$. When $z \in S$, by Lemma 1 applied to partition (D_S, S, C), z must see D_S and C. If z sees some vertex in $C \setminus \Omega$, we are done because z sees $D_x^W \cup D_y^W$ so we are in case 3a. Assume now that $N(z) \cap C \subseteq \Omega$, we prove that actually $N(z) \cap C = \Omega \cap C = T$, so we are in case 1. Assume there is $u \in T \setminus N(z)$. By Proposition 2, there must be a connected component D of $G - \Omega$ such that $z, u \in N(D)$. Since $u \in C$, this component D must be a subset of C, so $D \subseteq C \setminus \Omega$. Together with $z \in N(D)$, this contradicts the assumption $N(z) \cap C \subseteq \Omega$.

It remains to treat the case $z \in T$. Clearly $z \in C$ cannot see D_S because S separates C from D_S. We again take graph H, with vertex cover $(W \cap C) \cup \{x, y\}$, and apply Lemma 1 with partition $(D_x \cup \{x\}, T, D_y \cup \{y\})$. We deduce that z sees both $D_x^W \cup \{x\}$ and $D_y^W \cup \{y\}$. Assume that z does not see $D_x^W \cup D_y^W$. So $N(z) \cap C \setminus \Omega = \emptyset$ thus $N[z] \subseteq \Omega$. If Ω contains some vertex $u \notin N[z]$, no component of $G - \Omega$ can see both z and u (because $N(z) \subseteq \Omega$), contradicting Proposition 2. We conclude that either z sees $D_x^W \cup D_y^W$ (so satisfies condition 3b) or $\Omega = N[z]$ (thus we are in the second case of the Lemma). □

Theorem 2 *Every graph G contains $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ potential maximal cliques with active separators. Moreover the set of its potential maximal cliques with active separators can be listed in $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ time.*

Proof The number of potential maximal cliques with active separators satisfying the second condition of Lemma 2 is at most n, and they can all be listed in polynomial time by checking, for each vertex t, if $N[t]$ is a potential maximal clique.

For enumerating the potential maximal cliques with active separators satisfying the first condition of Lemma 2, we enumerate all minimal separators S using Theorem 1; there are at most $3^{\mathrm{vc}(G)}$ such sets. Then, for each $t \in S$ and each of the at most n components C of $G - S$ we check if $S \cup (C \cap N(t))$ is a potential maximal clique. Recall that testing if a vertex set is a potential maximal clique can be done in polynomial time by Corollary 1. Thus the whole process takes $\mathcal{O}^*(3^{\mathrm{vc}(G)})$ time, and this is also an upper bound on the number of listed objects.

It remains to enumerate the potential maximal cliques with active separators satisfying the third condition of Lemma 2. For this purpose, we "guess" the sets D_S^W D_x^W, D_y^W as in the Lemma and then we compute Ω. More formally, we enumerate all four-partitions $(D_S^W, D_x^W, D_y^W, \Omega^W)$ of W; there are exactly $4^{\mathrm{vc}(G)}$ such partitions. For each of them we let $\Omega^{\overline{W}}$ be the set of vertices $z \notin W$ satisfying conditions 3a or 3b of Lemma 2, and we test using Corollary 1 if $\Omega = \Omega^W \cup \Omega^{\overline{W}}$ is indeed a potential maximal clique. If so, we store Ω in a list of potential maximal cliques. By Lemma 2, this enumerates all potential maximal cliques of this type. The running time is $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ because for each four-partition $(D_S^W, D_x^W, D_y^W, \Omega^W)$ of W we performed a polynomial-time operation, computing the unique associated set Ω and testing whether it is a potential maximal clique. □

For counting and enumerating all potential maximal cliques of graph $G = (V, E)$, including the ones with no active separators, we apply the same ideas as in [6], based on the following statement.

Proposition 6 ([6]) *Let $G = (V, E)$ be a graph, let u be an arbitrary vertex of G and Ω be a potential maximal clique of G. Denote by $G - u$ the graph $G[V \setminus \{u\}]$. Then one of the following holds:*

1. *Ω has an active minimal separator S.*
2. *Ω is a potential maximal clique of $G - u$.*
3. *$\Omega \setminus \{u\}$ is a potential maximal clique of $G - u$.*
4. *$\Omega \setminus \{u\}$ is a minimal separator of G.*

Theorem 3 *Any graph G has $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ potential maximal cliques. Moreover the set of its potential maximal cliques can be listed in $\mathcal{O}^*(4^{\mathrm{vc}(G)})$ time.*

Proof Let $(v_1, \ldots, v_n)$ be an arbitrary ordering of the vertices of V. Denote by G_i the graph $G[\{v_1, \ldots, v_i\}]$ induced by the first i vertices, for all i, $1 \le i \le n$. Let $k = \mathrm{vc}(G)$. Note that for all i we have $\mathrm{vc}(G_i) \le k$. Actually, if W is a vertex cover of G, then $W_i = W \cap \{v_1, \ldots, v_i\}$ is a vertex cover of G_i. In particular, by Theorems 1 and 2, each G_i has at most 3^k minimal separators and $\mathcal{O}^*(4^k)$ potential maximal cliques with active separators.

For $i = 1$, graph G_1 has a unique potential maximal clique equal to $\{v_1\}$.

For each i from 2 to n, in increasing order, we compute the potential maximal cliques of G_i from those of G_{i-1} using Proposition 6. Observe that $G_{i-1} = G_i - v_i$. We initialize the set of potential maximal cliques of G_i with the ones having active separators. This can be done in $\mathcal{O}^*(4^k)$ time by Theorem 2. Then for each minimal separator S of G_i we check if $\Omega = S \cup \{v_i\}$ is a potential maximal clique of G_i and if so we add it to the set. This takes $\mathcal{O}^*(3^k)$ time by Theorem 1 and Corollary 1. Eventually, for each potential maximal clique Ω' of G_{i-1}, we test using Corollary 1 if Ω' (resp. $\Omega' \cup \{v_i\}$) is a potential maximal clique of G_i. If so, we add it to the set of potential maximal cliques of G_i. The running time of this last part is the number of potential maximal cliques of G_{i-1} times nm. Altogether, it takes $\mathcal{O}^*(4^k)$ time.

By Proposition 6, this algorithm covers alls cases and thus lists all potential maximal cliques of G_i. Hence for $i = n$ we obtain all potential maximal cliques of G, and they have been enumerated in $\mathcal{O}^*(4^k)$ time. □

4 Relations to Modular Width

A *module* of graph $G = (V, E)$ is a set of vertices W such that, for any vertex $x \in V \setminus W$, either $W \subseteq N(x)$ or W does not intersect $N(x)$. For the reader familiar with the modular decompositions of graphs, the modular width $\mathrm{mw}(G)$ of a graph G is the maximum size of a prime node in the modular decomposition tree. Equivalently, graph G is of modular width at most k if:

1. G has at most one vertex (the base case).
2. G is a disjoint union of graphs of modular width at most k.

3. G is a *join* of graphs of modular width at most k. I.e., G is obtained from a family of disjoint graphs of modular width at most k by taking the disjoint union and then adding all possible edges between these graphs.
4. The vertex set of G can be partitioned into $p \leq k$ modules $V_1, \ldots, V_p$ such that $G[V_i]$ is of modular width at most k, for all i, $1 \leq i \leq p$.

The modular width of a graph can be computed in linear time, using e.g. [30]. Moreover, this algorithm outputs the algebraic expression of G corresponding to the grammar above. The canonical way of defining such an expression is the *modular decomposition tree* of graph G, which is a rooted tree with labeled nodes of different types (see [30] for full details). If G has a unique vertex (first item above), the tree is reduced to a single node, of type *leaf*, labeled with the vertex itself. If G is not connected (second item), let $C_1, \ldots, C_p$ be the connected components of G. The modular decomposition of G is obtained by taking a root node of type *disjoint union*, whose sons are the roots of the modular decompositions of $G[C_1], \ldots, G[C_p]$. If $\overline{G}$ (the complement of graph G) is not connected, then we are in the case of the third item. Let $C_1, \ldots, C_p$ be the connected components of $\overline{G}$. The modular decomposition of G is obtained with a root node of type *join*, whose sons are the roots of the modular decompositions of $G[C_1], \ldots, G[C_p]$. When both G and $\overline{G}$ are connected (fourth item), there is a unique partition $(V_1, \ldots, V_p)$ of its vertex set V into maximal modules strictly contained in V. The modular decomposition of G has a root node of type *prime*, whose sons are the roots of the modular decompositions of $G[V_1], \ldots, G[V_p]$, in this order. The root is labeled with a graph $G[\{v_1, \ldots, v_p\}]$ obtained by choosing a vertex v_i in each V_i (observe that any such choice produces isomorphic labels). This label is a prime graph (hence the type of the root node). Equivalently, graph G has modular width at most k if all the prime nodes of its modular decomposition have at most k sons.

Let $G = (V, E)$ be a graph with vertex set $V = \{v_1, \ldots, v_k\}$ and let $M_i = (V_i, E_i)$ be a family of pairwise disjoint graphs, for all i, $1 \leq i \leq k$. Denote by H the graph obtained from G by replacing each vertex v_i by the module M_i. I.e., $H = (V_1 \cup \cdots \cup V_k, E_1 \cup \cdots \cup E_k \cup \{ab \mid a \in V_i, b \in V_j \text{ s.t. } v_i v_j \in E\})$. We say that graph H has been obtained from G by *expanding* each vertex v_i by the module M_i.

A vertex subset W of H is an *expansion* of vertex subset W_G of G if $W = \cup_{v_i \in W_G} V_i$. Given a vertex subset W of H, the *contraction* of W is $\{v_i \mid V_i \text{ intersects } W\}$.

Lemma 3 *Let S be a minimal y, z-separator of H, for $y, z \in V_i$. Then $S \cap V_i$ is a minimal separator of M_i and $S \setminus V_i = N_H(V_i)$.*

Proof Note that all vertices of $N_H(V_i)$ are in $N_H(y) \cap N_H(z)$, by construction of graph H and the fact that y and z are in the same module V_i. Therefore $N_H(V_i)$ must be contained in S. Let $S_i = S \cap V_i$. Since $H[V_i] = M_i$, we have that S_i separates z and y in graph M_i. Assume that S_i is not a minimal y, z-separator of M_i, so let $S'_i \subsetneq S_i$ be a minimal y, z-separator in graph M_i. We claim that $S'_i \cup N_H(V_i)$ is a y, z-separator in H. Indeed each y, z-path of H is either contained in V_i (in which case it intersects S'_i) or intersects $N_H(V_i)$. In both cases, it passes through $S'_i \cup N_H(V_i)$, which proves the claim. Since $S'_i \cup N_H(V_i)$ is a subset of S and S is a y, z-minimal separator of H, the only possibility is that $S = S'_i \cup N_H(V_i)$. This proves that $S \cap V_i$ is a minimal separator of M_i and $S \setminus V_i = N_H(V_i)$. □

Lemma 4 *Let S be a minimal separator of H. Assume that some V_i intersects S, but is not contained in S. Then V_i intersects all full components of $H - S$ associated to S. In particular $S \cap V_i$ is a minimal separator in M_i and $S \setminus V_i = N_H(V_i)$.*

Proof Let (x, t) be a pair of vertices with $x \in V_i \cap S$ and $t \in V_i \setminus S$. By Proposition 1, there are at least two full components of $H - S$, associated to S. Let C be one of them, not containing t. Let z be a neighbor of x in C, we prove that $z \in V_i$. If $z \notin V_i$, then $z \in N_H(V_i)$, and since V_i is a module in H we also have $z \in N_H(t)$. This contradicts the fact that t and z are in different components of $H - S$. It follows that $z \in V_i$. By applying the same arguments to the pair (x, z) instead of (x, t), it follows that V_i intersects each full component D of $H - S$ and moreover x has a neighbor in $D \cap V_i$.

By Proposition 1, S is a minimal y, z-separator in H, for some $y, z \in N_H(x) \cap V_i$. The rest follows by Lemma 3. □

Lemma 5 *Let S be a minimal separator of H. Then one of the following holds*

1. *S is the expansion of a minimal separator S_G of G.*
2. *There is $i \in \{1, \ldots, k\}$ such that $S \cap V_i$ is a minimal separator of M_i and $S \setminus V_i = N_H(V_i)$.*

Proof Assume there is a set V_i intersecting S but not contained in it. By Lemma 4, $S \cap V_i$ is a minimal separator of M_i and $S \setminus V_i = N_H(V_i)$. Hence we are in the second case of the Lemma.

Otherwise, for any V_i intersecting S, we have $V_i \subseteq S$. Thus S is the expansion of a vertex subset S_G of G, formed exactly by the vertices v_i of G such that V_i intersects S. Let C and D be two full components of $H - S$ associated to S and let $a \in C$, $b \in D$. Recall that, by Proposition 1, S is a minimal a, b-separator of H. Let V_k be the set containing a and V_l the set containing b. Consider first the possibility that $k = l$. Then, by Lemma 3, S satisfies the second condition of this lemma, for $i = k = l$ (This case may occur when M_k is disconnected and $S = N_H(V_k)$).

Finally, we consider the case $k \neq l$. We prove that S_G is a minimal v_k, v_l-separator of G. Consider a v_k, v_l path of G. If this path does not intersect S_G in G, then there is a path from a to b in $H - S$, obtained by replacing each vertex v_j of the path by some vertex of V_j (v_k and v_l are replaced by a and b respectively). This would contradict the fact that S separates a and b in H. Therefore S_G is indeed a v_k, v_l-separator in G. Assume that S_G is not minimal among the v_k, v_l-separators of G, and let $v_j \in S_G$ such that $S_G \setminus \{v_j\}$ separates v_k and v_l in G. We claim that $S \setminus V_j$ also separates a from b in H. By contradiction, assume there is a path from $a \in C \cap V_k$ to $b \in D \cap V_l$ in H, avoiding $S \setminus V_j$. By contracting, on this path, all vertices belonging to a same V_i into vertex v_i, we obtain a path (or a connected subgraph) joining v_k to v_l in G. This contradicts the fact that all such paths should intersect $S_G \setminus \{v_j\}$. Therefore S_G is a minimal separator of G. □

Lemma 5 provides an injective mapping from the set of minimal separators of H to the union of the sets of minimal separators of G and of the graphs M_i. Therefore we have:

Corollary 2 *The number of minimal separators of H is at most the number of minimal separators of G plus the number of minimal separators of each M_i.*

We now aim to prove a statement equivalent of Corollary 2, for potential maximal cliques instead of minimal separators.

Lemma 6 *Let Ω be a potential maximal clique of H, and let $\Omega_G = \{v_i \mid V_i$ intersects $\Omega\}$. Assume that Ω is the expansion of Ω_G, i.e. $\Omega = \cup_{v_i \in \Omega_G} V_i$. Then Ω_G is a potential maximal clique of G.*

Proof We prove that Ω_G satisfies, in graph G, the conditions of Proposition 2. For the first condition, let C_G be a component of $G - \Omega_G$ and let $S_G = N_G(C_G)$. Assume that S_G is not strictly contained in Ω_G, hence $S_G = \Omega_G$. Let C be the expansion of C_G in H and note that $N_H(C)$ is the expansion of $N_G(C_G)$, thus $N_H(C) = \Omega$. If C_G is formed by at least two vertices, since $G[C_G]$ is connected then so is $H[C]$. Therefore, in graph H, we have $N_H(C) = \Omega$ and C is a component of $H - \Omega$. But this contradicts the first condition of Proposition 2 applied to the potential maximal clique Ω of H. In the case that C_G is formed by a unique vertex v_k, its expansion $C = V_k$ might not induce a connected subset in H (if M_k is disconnected). But it is sufficient to consider a connected component V'_k of $H[V_k]$, and again this is also a component of $H - \Omega$ with the property that its neighborhood in H is the whole set Ω, contradicting Proposition 2 applied to Ω.

For the second condition of Proposition 2, let $v_j, v_k \in \Omega_G$ such that $v_j v_k$ is not an edge of G. Let $a \in V_j$ and $b \in V_k$. These vertices are non-adjacent in H, so by Proposition 2 applied to the potential maximal clique Ω of H there must be a component C of $H - \Omega$ seeing both a and b. Consider an a, b-path in $H[C \cup \{a, b\}]$. The contraction of this path contains a v_j, v_k-path in G, whose internal vertices are not in Ω_G. This proves that v_j and v_k are in the neighborhood of a same component of $G - \Omega_G$, thus Ω_G satisfies the second condition of Proposition 2. □

Lemma 7 *Let Ω be a potential maximal clique of H, and assume that there is some set V_i that intersects Ω but is not contained in Ω. Then $\Omega \cap V_i$ is a potential maximal clique of M_i and $\Omega \setminus V_i = N_H(V_i)$.*

Proof Let V_i be a vertex set that intersects Ω, but is not contained in Ω. We distinguish two cases.

Case 1 There is a minimal separator $S \subseteq \Omega$ of H, such that S intersects V_i.

By Lemma 4, $S \cap V_i$ is a minimal separator of M_i and V_i intersects all full components of $H - S$ associated to S. Let C be the unique component of $H - S$ intersecting Ω; recall that it exists and moreover it is full w.r.t. S, by Proposition 3. Then, by Lemma 4, C also intersects V_i. Also by Lemma 4, $S \setminus V_i = N_H(V_i)$. We claim that actually $C \subseteq V_i$ and C is also a full component of $M_i - S_i$. Recall that $S \setminus V_i = N_H(V_i)$ separates in graph H the vertices of V_i from the rest of the graph. Since C intersects V_i, $H[C]$ is connected and $N_H(V_i)$ separates V_i from all other vertices, we must have $C \subseteq V_i$. Since $H[C]$ is connected, so is $M_i[C]$, thus C is contained in some component of $M_i - S_i$. But each such component is also a component of $H - S$, hence C is both a component of $H - S$ and of $M_i - S_i$. In particular $\Omega \cap C \subseteq V_i$.

It remains to prove that $\Omega_i = \Omega \cap V_i$ is a potential maximal clique of M_i. By the above observations, we also have $\Omega_i = \Omega \setminus N_H(V_i)$. We show that Ω_i satisfies, in graph M_i, the conditions of Proposition 2. Let D be a component of $M_i - \Omega_i$. Observe that D is also a component of $H - \Omega$ and let $T = N_{M_i}(D)$. Either D is a component of $M_i - \Omega_i$ disjoint from C, or it is contained in C. In the former case, T is a subset of S_i, hence T is a strict subset of Ω_i (since S_i is itself a strict subset of Ω_i by Proposition 2 applied to potential maximal clique Ω of H). In the latter case, if $T = \Omega_i$, note that $N_H(D) = \Omega$ because $\Omega \setminus V_i = N_H(V_i)$ is also contained in the neighborhood of D in H. This contradicts Proposition 2 applied to potential maximal clique Ω of H.

For the second condition, let $x, y \in \Omega_i$, non-adjacent in M_i. Then there is a component F of $H - \Omega$ seeing, in graph H, both x and y (by Proposition 2 applied to Ω). Since this component sees V_i, it must be contained in V_i. So F is also a component of $M_i - \Omega_i$ seeing both x and y in M_i, which concludes our proof for this case.

Case 2 There is no minimal separator $S \subseteq \Omega$ of H, intersecting V_i.

Let us prove that, in this case, for any $x \in \Omega_i$ we have that $\Omega = N_H[x]$. By Proposition 3, if x has a neighbor outside Ω, hence in some component D of $H - \Omega$, then $N_H(D)$ is a minimal separator of H containing x — a contradiction. Therefore $N_H[x] \subseteq \Omega$. If there is $y \in \Omega \setminus N_H[x]$, then by the same proposition, x and y must see a same component of $H - \Omega$, contradicting the fact that $N_H(x) \subseteq \Omega$. We deduce that $\Omega = N_H[x]$.

Since this holds for each vertex of Ω_i, we have that Ω_i is a clique in H (thus in M_i), and the vertices of Ω_i cannot have neighbors in $V_i \setminus \Omega_i$. Therefore Ω_i is a clique and a connected component of M_i. By Proposition 3, it is a potential maximal clique of M_i. The fact that $\Omega = N_H[x]$ also implies that $\Omega \setminus V_i = N_H(V_i)$, which concludes our proof. □

From Lemmata 6 and 7, we directly deduce:

Lemma 8 *Let Ω be a potential maximal clique of H. One of the following holds*

1. *Ω is the expansion of a potential maximal clique Ω_G of G.*
2. *There is some $i \in \{1, \dots, k\}$ such that $\Omega \cap V_i$ is a potential maximal clique of M_i and $\Omega \setminus V_i = N_H(V_i)$.*

The previous lemma provides an injective mapping from the set of potential maximal cliques of H to the union of the sets of potential maximal cliques of G and of the graphs M_i. Therefore we have:

Corollary 3 *The number of potential maximal cliques of H is at most the number of potential maximal cliques of G plus the number of potential maximal cliques of each M_i.*

The following proposition bounds the number of minimal separators and potential maximal cliques of arbitrary graphs with respect to n.

Proposition 7 ([19,20]) *Every n-vertex graph has $\mathcal{O}(\rho^n)$ minimal separators, where $\rho < 1.6181$ is the golden ratio, and $\mathcal{O}(1.7347^n)$ potential maximal cliques. Moreover, these objects can be enumerated within the same running times.*

We can now prove the main result of this section.

Theorem 4 *For any graph $G = (V, E)$, the number of its minimal separators is $\mathcal{O}(n \cdot \rho^{\mathrm{mw}(G)})$ where $\rho < 1.6181$ is the golden ratio. The number of its potential maximal* *cliques is* $\mathcal{O}(n \cdot 1.7347^{\mathrm{mw}(G)})$*. Moreover, the minimal separators and the potential maximal cliques can be enumerated in time* $\mathcal{O}^*(1.6181^{\mathrm{mw}(G)})$ *and* $\mathcal{O}^*(1.7347^{\mathrm{mw}(G)})$ *time respectively.*

Proof Let $k = \mathrm{mw}(G)$. By definition of modular width, there is a modular decomposition tree of graph G, each node corresponding to a leaf, a disjoint union, a join or a decomposition into at most k modules. The leaves of the decomposition tree are disjoint graphs with a single vertex, thus these vertices form a partition of V. In particular, there are at most n leaves and, since each internal node is of degree at least two, there are $O(n)$ nodes in the decomposition tree. For each node *Node*, let $G(Node)$ be the graph associated to the subtree rooted at *Node*. I.e., $G(Node)$ is the graph whose modular decomposition is the subtree rooted at *Node*; it is also subgraph of G induced by the vertices of G mapped on leaves of the subtree rooted at *Node*. We prove that $G(Node)$ has $\mathcal{O}(n(Node) \cdot \rho^k)$ minimal separators and $\mathcal{O}(n(Node) \cdot 1.7347^k)$ potential maximal clique, where $n(Node)$ is the number of nodes of the subtree rooted at *Node*. We proceed by induction from bottom to top. The statement is clear when *Node* is a leaf.

Let *Node* be an internal node $Node_1, Node_2, \ldots, Node_p$ be its children in the tree. Graph $G(Node)$ is the expansion of some graph $G'(Node)$ by replacing the i-th vertex with module $G(Node_i)$. If *Node* is a *join* node, then $G'(Node)$ is a clique. When *Node* is a *disjoint union* node, graph $G'(Node)$ is an independent set, and in the last case $G'(Node)$ is a graph of at most k vertices. In all cases, by Proposition 7 graph $G'(Node)$ has $\mathcal{O}(\rho^k)$ minimal separators. Thus $G(Node)$ has at most $\mathcal{O}(\rho^k)$ more minimal separators than all graphs $G(Node_i)$ taken together, which completes our proof for minimal separators.

Concerning potential maximal cliques, when $G'(Node)$ is a clique it has exactly one potential maximal clique, and when $G'(Node)$ is of size at most k is has $\mathcal{O}(1.7347^k)$ potential maximal cliques. We must be more careful in the case when $G'(Node)$ is an independent set (i.e., *Node* is a disjoint union node), since in this case it has p potential maximal cliques, one for each vertex, and p can be as large as n. Consider a potential maximal clique Ω of $G(Node)$ corresponding to an expansion of vertices of $G'(Node)$ (see Lemma 8). It follows that this potential maximal clique is exactly the vertex set of some $G(Node_i)$, for a child $Node_i$ of *Node*. By construction this vertex set is disconnected from the rest of $G(Node)$, and by Proposition 2 the only possibility is that this vertex set induces a clique in $G(Node)$. But in this case Ω is also a potential maximal clique of $G(Node_i)$. This proves that, when *Node* is of type disjoint union, $G(Node)$ has no more potential maximal cliques than the sum of the numbers of potential maximal cliques of all $G(Node_i)$, $1 \leq i \leq p$. We conclude that the whole graph G has $\mathcal{O}(n \cdot 1.7347^k)$ potential maximal cliques. All our arguments are constructive and can be turned directly into enumeration algorithms for these objects. □

5 Applications

The *treewidth* of graph $G = (V, E)$, denoted $\mathrm{tw}(G)$, is the minimum number k such that G has a triangulation $H = (V, E')$ of clique size at most $k+1$. The *minimum fill-in* of G is the minimum size of F, over all (minimal) triangulations $H = (V, E \cup F)$ of G. The *treelength* of G is the minimum k such that there exists a minimal triangulation H, with the property that any two vertices adjacent in H are at distance at most k in graph G.

Proposition 8 *Let Π_G denote the set of potential maximal cliques of graph G. The following problems are solvable in $\mathcal{O}^*(|\Pi_G|)$ time, when Π_G is given in the input:* (Weighted) Treewidth *[4,16],* (Weighted) Minimum Fill- In *[16,23],* Treelength *[26].*

Let us also recall the Max Induced Subgraph of tw $\leq t$ satisfiying φ problem where, for a fixed integer t and a fixed $CMSO_2$ formula φ, the goal is to find a pair of vertex subsets $X \subseteq F \subseteq V$ such that $\text{tw}(G[F]) \leq t$, $(G[F], X)$ models φ and X is of maximum size.

Proposition 9 ([18]) *For any fixed integer $t > 0$ and any fixed* $CMSO_2$ *formula φ, problem* Max Induced Subgraph of tw $\leq t$ satisfiying φ *is solvable in $\mathcal{O}(|\Pi_G| \cdot n^{t+4})$ time, when Π_G is given in the input.*

Pipelined with Theorems 3 and 4, we deduce:

Theorem 5 *For an input graph G, the problems* Max Induced Subgraph of-tw $\leq t$ satisfiying φ, (Weighted) Treewidth, (Weighted) Minimum Fill- In *and* Treelength *are solvable in time $\mathcal{O}^*(4^{\text{vc}(G)})$, and in time $\mathcal{O}^*(1.7347^{\text{mw}(G)})$.*

We recall that problem Max Induced Subgraph of tw $\leq t$ satisfiying φ generalizes many classical problems, e.g., Maximum Independent Set, Maximum Induced Forest, Longest Induced Path, Maximum Induced Matching, Independent Cycle Packing, k- in- a- Path, k- in- a- Tree, Maximum Induced Subgraph With a Forbidden Planar Minor. More examples of particular cases are given in "Appendix 1" (see also [18]).

The polynomial factors hidden by the $\mathcal{O}^*$ notation depend on the problem and on the parameter, they are typically between n^5 to n^7.

6 Treedepth

In [13], Deogun et al. give the following formula for computing the treedepth of a graph. Note that, for technical reasons, the formula uses a super-set Δ_G^+ of the set of all minimal separators of G.

Proposition 10 ([13]) *Let G be a graph and Δ_G^+ a set of vertex subsets, containing all minimal separators of G. Then*

$$\text{td}(G) = \min_{S \in \Delta_G^+} \left(|S| + \max_{C} \text{td}(G[C]) \right)$$

where the maximum is taken over all components C of $G - S$.

Note that if G is a complete graph, then its treedepth is the number of vertices of G.

Following [13], we recursively define a *piece* D of $G = (V, E)$ as a vertex subset such that $D = V$, or there exists a larger piece C of G and a minimal separator S of $G[C]$ such that D is a connected component of $G[C] - S$.

Let us say that (C, S) is a *piece-separator pair* of G if C is a piece and S is a minimal separator of $G[C]$. Proposition 10 leads in [13] to a natural dynamic programming algorithm for computing the treedepth. Let us restate it for our convenience.

Lemma 9 *Given as input a graph G together with the set PS of all its piece-separator pairs (or a superset of this set), the treedepth of G can be computing in $\mathcal{O}^*(|PS|)$ time.*

Proof Let $Pieces$ be the set of all sets C such that $(C, S) \in PS$ (hence $Pieces$ is a superset of all pieces). Assume that the elements of $Pieces$ are ordered by inclusion, or simply by size, using bucket sort. The goal is to compute, in this order, a quantity that corresponds to the treedepth of $G[C]$, if C is really a piece. (Technically, we may encounter sets C that do not correspond to pieces, because when we will generate the sets C we will not be able to test that we only generate pieces). For the minimal sets $C \in Pieces$, we set $\text{td}[C] \leftarrow |C|$. Then for each $C \in Pieces$, sorted by inclusion, for each S such that $(C, S) \in PS$, we check that all components D of $G[C] - S$ belong to $Pieces$. If this is not the case, then C is not a piece. Otherwise, we compute $|S| + \sum_D \text{td}[D]$ (the sum is taken over all components D), and we set $\text{td}[C]$ to the minimum among these values, over all such S. (If no such S exist, we can set $\text{td}[C] \leftarrow |C|$). By Proposition 10, at the end of the algorithm, $\text{td}(G) = \text{td}[V]$. □

Following the same ideas as in Theorem 1, we can list the piece-separator pairs of G in a running time which is single-exponential in its vertex cover.

Lemma 10 *Let W be a vertex cover of graph $G = (V, E)$.*

For each piece C of G, either C is a singleton or there is a three-partition (C^W, T^W, R^W) of W such that $C \cap W = C^W$ and $C \setminus W$ is formed by the vertices $x \in V \setminus W$ such that $N(x)$ is contained in $C^W \cup T^W$ and intersects C^W.

Proof The proof goes by induction, from larger to smaller pieces. The base case corresponds to piece $C = V$. The condition holds, for the partition $(W, \emptyset, \emptyset)$.

Let now C be a piece obtained as a component of $G[F] - Q$, for some larger piece F and some minimal separator Q of $G[F]$. We may assume w.l.o.g. that C is a full component associated to Q in $G[F]$. Indeed, if this is not the case, then $Q' = N_{G[F]}(C)$ is also a minimal separator of $G[F]$ (C is one of the full components associated to Q' in $G[F]$; another one can be obtained from a full component D associated to Q, by taking the component of $G[F] - Q'$ containing D, see Proposition 1). Thus we can replace Q by Q'.

By induction hypothesis, there is a partition (F^W, T_F^W, R_F^W) such that $F \cap W = F^W$ and for any vertex $y \in F \setminus W$, y is in F if and only if $N_G(y)$ intersects F^W and is contained in $F^W \cup T_F^W$. Consider the partition (C, S, D_2) of F and apply Lemma 1 to $G[F]$, this partition and the vertex cover $W' = W \cap F$ of $G[F]$. Let $C^W = C \cap W'$,

$S^W = S \cap W'$ and $D_2^W = D_2 \cap W'$ (we obtain the same intersections if we replace W' by W). Denote $T^W = T_F^W \cup S^W$, and $R^W = R_F^W \cup D_2^W$. Observe that (C^W, T^W, R^W) is a three-partition of W. It remains to prove that, if C has more than one vertex, then this three-partition of W satisfies the condition of the lemma.

Let x be a vertex of $C \setminus W$, in particular $x \in F$. Since C is not a singleton, x has at least one neighbor in $C \cap W = C^W$, in graph G. Since $x \in F$, its neighborhood $N_G(x)$ is contained in $F \cup T_F^W$ by induction hypothesis. Now if $N_G(x)$ intersects D_2^W, we would have that $N_{G[F]}(x)$ intersects both C^W and D_2^W, contradicting (by Lemma 1) the fact that $x \in C$. It remains that $N_G(x)$ is contained in $C^W \cup T^W$.

Conversely, let $x \in V \setminus W$ such that $N_G(x)$ intersects C^W and is contained in $C^W \cup T^F$. Note that $x \in F$, by induction hypothesis. We cannot have $x \in S$ (its neighborhood would intersect D_2^W, hence R^W, by Lemma 1 applied on (C, S, D_2) in $G[F]$), nor $x \in D_2$ (because S separates C from D_2 in $G[F]$, so $N_G(D_2)$ cannot intersect C). It remains that $x \in C$, concluding the proof of the lemma. □

Theorem 6 *A graph G has at most $n + 3^{\mathrm{vc}(G)}$ pieces and at most $n + 5^{\mathrm{vc}(G)}$ piece-separator pairs. Moreover, the piece-separator pairs of G can be listed in $\mathcal{O}^*(5^{\mathrm{vc}(G)})$ time.*

Proof The fact that G has at most $n + 3^{\mathrm{vc}(G)}$ pieces is a straightforward consequence of Lemma 10: each piece C is either a single vertex, or is completely determined by some three-partition (C^W, T^W, R^W) of a minimum vertex cover W. We point out that the singleton condition of Lemma 10 is crucial. Indeed, if G is a star, its minimum vertex cover is of size one, and each of the other $n-1$ vertices is a piece formed by a single vertex.

Let us enumerate and upper bound the number of piece-separator pairs (C, S). There are at most n such pairs where C is a singleton, so assume that C has at least two vertices. For each such pair, let (C^W, T^W, R^W) be a partition of a minimum vertex cover W, like in Lemma 10. Since S is a minimal separator of $G[C]$, there is (by Lemma 1) a partition (D_1^W, S^W, D_2^W) of C^W such that $S \setminus W$ corresponds to vertices of $x \in C \setminus W$ whose neighborhood in $G[C]$ intersects both D_1^W and D_2^W. Conversely, by Lemmata 10 and 1, the pair (C, S) is completely determined by the five-partition $(D_1^W, S^W, D_2^W, T^W, R^W)$ of W. Indeed, given this five-partition, we set $C^W = D_1^W \cup S^W \cup D_2^W$. By Lemma 10, we can fix C as the vertex set formed by C^W and the vertices $z \in V \setminus W$ such that $N(x)$ intersects C^W and is contained in $C^W \cup T^W$. By Lemma 1 applied to $G[C]$, S is formed by S^W and the vertices $y \in C \setminus W$ seeing both D_1^W and D_2^W. Hence there are at most $5^{\mathrm{vc}(G)}$ such piece-separator pairs, which proves the upper bound.

The enumeration algorithm firstly enumerates all pairs of type $(C, \emptyset)$ where $|C| = 1$. Then it enumerates the $5^{\mathrm{vc}(G)}$ five-partitions $(D_1^W, S^W, D_2^W, T^W, R^W)$ of W, and for each of them it constructs as described before the unique associated pair (C, S). Therefore the algorithm enumerates a superset of the piece-separator pairs of G within the required running time. □

A similar result can be obtained using parameter modular width. As in Sect. 4, let $G = (V, E)$ be a graph with vertex set $V = \{v_1, \dots, v_k\}$ and let $M_i = (V_i, E_i)$ be a family of pairwise disjoint graphs, for all i, $1 \leq i \leq k$. We denote by H the graph obtained from G by expanding each vertex v_i by the module M_i.

Lemma 11 *Let S be a minimal separator of H and C be a connected component of $H - S$. Then one of the following holds:*

1. *C is the expansion of a vertex subset C_G of G.*
2. *There is $i \in \{1, \ldots, k\}$ and a minimal separator S_i of $M_i = H[V_i]$ such that C is a connected component of $M_i - S_i$.*

In particular, each piece of H is an expansion of a vertex subset of G, or a piece of some module M_i.

Proof The two conditions are a straightforward consequence of Lemma 5. Recall that, by this Lemma, S is the expansion of a minimal separator S_G of G (in which case the first condition holds), or there is some i and a minimal separator S_i of M_i such that $S = S_i \cup N_H(V_i)$. In the latter case, the component C is either contained in V_i (and the second condition holds), or does not intersect V_i, so it is the expansion of some component of $G - N_G(v_i)$ (implying the first condition of the lemma).

For the last part, we use the recursive definition of pieces. Consider a piece of $H[C]$, for some component C of $H - S$. If C is a component of $M_i - S_i$ for some i, then pieces of $H[C]$ are also pieces of $M_i[C]$ and hence of M_i. If C is an expansion of a vertex subset C_G of G, then $H[C]$ is an expansion of $H' = G[C_G]$ and we recursively apply the same argument on H'. □

Theorem 7 *For any graph G, the number of its pieces is $\mathcal{O}^*(2^{\mathrm{mw}(G)})$, and a superset of its piece-separator pairs can be listed in time $\mathcal{O}^*(3.2361^{\mathrm{mw}(G)})$.*

Proof Let us count the pieces of G. Consider the modular-decomposition tree of G, like in the proof of Theorem 4. For each node *Node* of the decomposition (corresponding to a leaf, a disjoint union, a join or a decomposition into at most $\mathrm{mw}(G)$ modules), let $G(\textit{Node})$ be the graph whose corresponding decomposition tree is the subtree rooted at *Node*, and $G'(\textit{Node})$ be the prime, complete or independent graph corresponding to node *Node*. By Lemma 11, the pieces of $G(\textit{Node})$ are either pieces of $G(\textit{Node}_i)$ for some child $\textit{Node}_i$ of *Node*, or correspond to an expansion of $G'(\textit{Node})$. If $G'(\textit{Node})$ is prime, there are at most $2^{\mathrm{mw}(G)}$ such expansions. Also observe that if $G'(\textit{Node})$ is independent then the only pieces of $G(\textit{Node})$ that are expansions of $G'(\textit{Node})$ correspond to single vertices of $G'(\textit{Node})$. Hence, there are at most n such pieces. In the case when $G'(\textit{Node})$ is a complete graph, the only possible piece of $G(\textit{Node})$ that is an expansion of $G'(\textit{Node})$ is the whole graph $G(\textit{Node})$, hence this piece is unique. Altogether, G has $\mathcal{O}^*(2^{\mathrm{mw}(G)})$ pieces.

For each piece C of G, there are $\mathcal{O}^*(\rho^{\mathrm{mw}(G)})$ minimal separators in $G[C]$, where ρ is the golden ratio, by Theorem 4 and the fact that the modular width of $G[C]$ is at most $\mathrm{mw}(G)$. Hence the number of piece-separator pairs of G is $\mathcal{O}^*((2 \cdot \rho)^{\mathrm{mw}(G)})$, thus $\mathcal{O}^*(3.2361^{\mathrm{mw}(G)})$. The counting arguments can be transformed into an enumeration algorithm for a superset of all possible such pairs, within the same running time bounds. □

From Lemma 9 and Theorems 6 and 7 we deduce:

Theorem 8 *The treedepth of a graph G is computable in time* $\mathcal{O}^*(5^{\mathrm{vc}(G)})$ *and in time* $\mathcal{O}^*(3.2361^{\mathrm{mw}(G)})$.

7 Conclusion

We have proved single exponential upper bounds for the number of minimal separators and the number of potential maximal cliques of graphs, with respect to parameters vertex cover and modular width. As a consequence, we provide a unified framework for solving several classical optimization problems in single-exponential time with respect to these parameters (e.g., Treewidth, Minimum Fill- In, Longest Induced Path, Independent Cycle Packing, see also Theorem 5 and "Appendix 1" for more applications). Some of these results have been proved before, but using ad-hoc algorithms (for example Treewidth parameterized by vertex cover [7] and modular width [3]), others are new.

A natural question is whether the technique can be extended to other natural graph parameters, by obtaining upper bounds of type $\mathcal{O}^*(f(k))$ on the number of potential maximal cliques, for any function f (here k is the parameter). To the best of our knowledge, prior to our work this question has not been investigated for any relevant parameter except for n, the number of vertices of the graph. We point out that for parameters like feedback vertex set, clique-width or maximum leaf spanning tree, one cannot obtain such upper bounds. A counterexample is provided by the graph $W_{p,q}$, formed by p disjoint paths of q vertices plus two vertices u and v seeing the left, respectively right ends of the paths (similar to the watermelon graph of Fig. 1). Indeed this graph has feedback vertex set 1, a maximum leaf spanning tree with p leaves and a clique-width of no more than $2p+1$, but it has roughly $p^{n/p}$ minimal u, v-separators. Skodinis [29] observes that, if we chose as parameter d the maximum degree of the complement graph, then the number of minimal separators is bounded by $\mathcal{O}^*(2^{\mathcal{O}(d)})$. The argument is that each a, b-minimal separator contains all common neighbors of a and b, hence all but $2d$ vertices of the graph. A similar bound holds for the number of potential maximal cliques.

A different extension of this technique was recently considered by Liedloff et al. [25], with different parameters including feedback vertex set. They can solve the problem Max Induced Subgraph of tw $\leq t$ satisfiying φ in FPT time, but unfortunately the running time is not single exponential, and their result does not extend to problems like Treewidth or Treedepth.

Finally, we point out that our bounds on the number of potential maximal cliques w.r.t. vertex cover and to modular width do not seem to be tight. Any improvement on these bounds and of the enumeration algorithm of potential maximal cliques would imply faster algorithms for the problems mentioned in Sect. 5.

Appendix 1: More Applications

We give in this Appendix several problems that are all known to be particular cases of Max Induced Subgraph of tw $\leq t$ satisfiying φ (see [18] proofs and more applications). Proposition 9 also extends to the weighted version and the annotated version of the problems (in the annotated version, a fixed vertex subset must be part of the solution F).

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 and a finite set of graphs $\mathcal{F}_{plan}$ containing a planar graph we define he following generic problem.

> MAXIMUM IND. 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 $\mathcal{F}_{plan}$.

Even the special case with $\ell = 0$, this problem and its complementary version called the MINIMUM $\mathcal{F}$- DELETION, encompass many different problems.

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 $\mathcal{G}(t, \varphi)$.

As natural sub cases studied in the literature we can cite INDEPENDENT TRIANGLE PACKING or INDEPENDENT CYCLE PACKING.

The next problem is an example of *annotated version* of optimization problem MAX INDUCED SUBGRAPH OF tw $\leq t$ SATISFIYING φ.

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

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.

Appendix 2: Monadic Second-Order Logic

We use Counting Monadic Second Order Logic ($CMSO_2$), an extension of MSO_2, as a basic tool to express properties of vertex and edge sets in graphs.

The syntax of Monadic Second Order Logic (MSO_2) of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges, sets of vertices, and sets of edges, the quantifiers $\forall$, $\exists$ that can be applied to these variables, and the following five binary relations:

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

The MSO_1 is a restriction of MSO_2 in which one cannot use edge set variables (in particular the incidence relation becomes unnecessary). For example HAMILTONICITY is expressible in MSO_2 but not in MSO_1.

In addition to the usual features of monadic second-order logic, if we have atomic sentences testing whether the cardinality of a set is equal to q modulo r, where q and r are integers such that $0 \leq q < r$ and $r \geq 2$, then this extension of the MSO_2 (resp. MSO_1) is called the *counting monadic second-order logic* $CMSO_2$ (resp. $CMSO_1$). So essentially $CMSO_2$ (resp. $CMSO_1$) is MSO_2 (resp. MSO_1) with the following atomic sentence for a set S:

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

We refer to [1,9] and the book of Courcelle and Engelfriet [10] for a detailed introduction on different types of logic.

References

1. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms **12**, 308–340 (1991)
2. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**, 1305–1317 (1996)

3. Bodlaender, H.L., Fomin, F.V.: Tree decompositions with small cost. Discrete Appl. Math. **145**, 143–154 (2005)
4. Bodlaender, H.L., Rotics, U.: Computing the treewidth and the minimum fill-in with the modular decomposition. Algorithmica **36**, 375–408 (2003)
5. Bouchitté, V., Todinca, I.: Treewidth and minimum fill-in: Grouping the minimal separators. SIAM J. Comput. **31**, 212–232 (2001)
6. Bouchitté, V., Todinca, I.: Listing all potential maximal cliques of a graph. Theor. Comput. Sci. **276**, 17–32 (2002)
7. Chapelle, M., Liedloff, M., Todinca, I., and Villanger, Y.: Treewidth and pathwidth parameterized by the vertex cover number. In: WADS, Dehne, F., Solis-Oba, R., Sack, J.-R. (eds.) vol. 8037 of Lecture Notes in Computer Science, pp. 232–243. Springer, Berlin (2013)
8. Chen, J., Kanj, I.A., Xia, G.: Improved upper bounds for vertex cover. Theor. Comput. Sci. **411**, 3736–3756 (2010)
9. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Inf. Comput. **85**, 12–75 (1990)
10. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic. Cambridge University Press, Cambridge (2012)
11. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**, 125–150 (2000)
12. Cygan, M., Lokshtanov, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: On cutwidth parameterized by vertex cover. Algorithmica **68**, 940–953 (2014)
13. Deogun, J.S., Kloks, T., Kratsch, D., Müller, H.: On the vertex ranking problem for trapezoid, circular-arc and other graphs. Discrete Appl. Math. **98**, 39–63 (1999)
14. Fellows, M.R., Lokshtanov, D., Misra, N., Rosamond, F.A., Saurabh, S.: Graph layout problems parameterized by vertex cover. In: ISAAC, Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) vol. 5369 of Lecture Notes in Computer Science, pp. 294–305. Springer, Berlin (2008)
15. Fomin, F.V., Giannopoulou, A.C., Pilipczuk, M.: Computing tree-depth faster than 2^n. Algorithmica **73**, 202–216 (2015)
16. Fomin, F.V., Kratsch, D., Todinca, I., Villanger, Y.: Exact algorithms for treewidth and minimum fill-in. SIAM J. Comput. **38**, 1058–1079 (2008)
17. Fomin, F.V., Liedloff, M., Montealegre-Barba, P., and Todinca, I.: Algorithms parameterized by vertex cover and modular width, through potential maximal cliques. In: Ravi, R., L. Gørtz, I. (eds.) Algorithm Theory—SWAT 2014—14th Scandinavian Symposium and Workshops, Copenhagen, Denmark, July 2-4, 2014. Proceedings, vol. 8503 of Lecture Notes in Computer Science, pp. 182–193. Springer, Berlin (2014)
18. Fomin, F.V., Todinca, I., Villanger, Y.: Large induced subgraphs via triangulations and CMSO. SIAM J. Comput. **44**, 54–87 (2015)
19. Fomin, F.V., Villanger, Y.: Finding induced subgraphs via minimal triangulations. In: STACS, Marion, J.-Y., Schwentick, T. (eds.) vol. 5 of LIPIcs, Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, pp. 383–394 (2010)
20. Fomin, F.V., Villanger, Y.: Treewidth computation and extremal combinatorics. Combinatorica **32**, 289–308 (2012)
21. Frick, M., Grohe, M.: The complexity of first-order and monadic second-order logic revisited. Ann. Pure Appl. Logic **130**, 3–31 (2004)
22. Gajarský, J., Lampis, M., Ordyniak, S.: Parameterized algorithms for modular-width. In: Parameterized and Exact Computation—8th International Symposium, IPEC 2013, Lecture Notes in Computer Science, pp. 163–176 (2013)
23. Gysel, R.: Minimal triangulation algorithms for perfect phylogeny problems. In: Dediu, A. H., Martín-Vide, C., Sierra-Rodríguez, J. L., Truthe, B. (eds.) Language and Automata Theory and Applications—8th International Conference, LATA 2014, Madrid, Spain, March 10–14, 2014. Proceedings, vol. 8370 of Lecture Notes in Computer Science, pp. 421–432. Springer, Berlin (2014)
24. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. Algorithmica **64**, 19–37 (2012)
25. Liedloff, M., Montealegre-Barba, P., Todinca, I.: Beyond classes of graphs with "few" minimal separators: FPT results through potential maximal cliques. In: Graph-Theoretic Concepts in Computer Science, 41st International Workshop, Proceedings of WG 2015, Munich, (2015) (**To appear**)
26. Lokshtanov, D.: On the complexity of computing treelength. Discrete Appl. Math. **158**, 820–827 (2010)

27. Nešetřil, J., de Mendez, P.O.: Sparsity—Graphs, Structures, and Algorithms, vol. 28 of Algorithms and combinatorics, Springer, 2012.
28. Reidl, F., Rossmanith, P., Villaamil, F.S., Sikdar, S.: A faster parameterized algorithm for treedepth. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) Automata, Languages, and Programming—41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I, vol. 8572 of Lecture Notes in Computer Science, pp. 931–942. Springer, Berlin (2014)
29. Skodinis, K.: Efficient analysis of graphs with small minimal separators. In: Widmayer, P., Neyer, G., Eidenbenz, S. (eds.) Graph-Theoretic Concepts in Computer Science, 25th International Workshop, WG '99, Ascona, Switzerland, June 17–19, 1999, Proceedings, vol. 1665 of Lecture Notes in Computer Science, pp. 155–166. Springer, Berlin (1999)
30. Tedder, M., Corneil, D.G., Habib, M., Paul, C.: Simpler linear-time modular decomposition via recursive factorizing permutations. In: ICALP (1) Automata, Languages and Programming, 35th International Colloquium, Lecture Notes in Computer Science, pp. 634–645 (2008)