

Algorithmic complexity of finding cross-cycles in flag complexes

Michał Adamaszek* and Juraj Stacho†

Mathematics Institute and DIMAP, University of Warwick, Coventry CV4 7AL, United Kingdom
(e-mail: aszek@mimuw.edu.pl, stacho@cs.toronto.edu)

Abstract

A cross-cycle in a flag simplicial complex K is an induced subcomplex that is isomorphic to the boundary of a cross-polytope and that contains a maximal face of K . A cross-cycle is the most efficient way to define a non-zero class in the homology of K . For an independence complex of a graph G , a cross-cycle is equivalent to an induced matching containing a maximal independent set of G .

We study the complexity of finding cross-cycles in independence complexes. We show that in general this problem is NP-complete when input is a graph whose independence complex we consider. This allows us to study special cases. Unfortunately, not a lot has been done in this direction besides the recent polynomial time algorithm for forests [16].

In this contribution, we focus on the more general class of chordal graphs. This is a natural choice for the problem as the independence complexes of chordal graphs are quite well understood, namely they are wedges of spheres up to homotopy, and any wedge of spheres can be realized as the independence complex of a chordal graph, up to homotopy.

As our main result, we present a polynomial time algorithm for detecting a cross-cycle in the independence complex of a chordal graph. Our algorithm is based on the geometric intersection representation of chordal graphs and has an efficient implementation.

We further prove that for chordal graphs cross-cycles detect all of homology of the independence complex. As a corollary, we obtain polynomial time algorithms for such topological properties as contractibility or simple-connectedness of independence complexes of chordal graphs. These problems are undecidable for general independence complexes.

We conclude with a discussion of some related cases and open problems.

1 Introduction

The purpose of this paper is to investigate the algorithmic complexity of detecting certain non-trivial homology classes in flag simplicial complexes and to explore the consequences for deciding their topological properties. There is a certain efficient construction that defines a class in the homology of a flag complex and, at the same time, delivers a proof that this class is non-trivial. We describe it now. Let S^0 be the zero-sphere, that is the simplicial complex consisting of two disjoint vertices. The k -fold join $O_k = S^0 * \cdots * S^0$ is a complex with $2k$ vertices, combinatorially equivalent to the boundary of the cross-polytope and homeomorphic to the $(k-1)$ -sphere.

Definition. A cross-cycle (of size k) in a flag simplicial complex K is an induced subcomplex of K isomorphic to O_k and such that it contains a maximal face of K .

A cross-cycle is an embedded sphere $S^{k-1} \subseteq K$, hence it defines a homology class in $\tilde{H}_{k-1}(K)$. This class must be non-zero as its representing cycle contains a maximal face, so

*Research of MA supported by the Centre for Discrete Mathematics and its Applications (DIMAP), EPSRC award EP/D063191/1.

†JS gratefully acknowledges support from EPSRC, grant EP/I01795X/1.

it cannot be hit by a differential (see Section 3 for details). The name cross-cycle refers to the fact that the sphere is isomorphic to the boundary of the cross-polytope.

Cross-cycles have been used to construct homology classes in a number of contexts [1, 3, 13, 20]. They appear as the main contribution to the homology of the clique complexes of random geometric graphs [15]. Moreover, they are the minimal models, in the sense that for a flag complex K every non-zero homology class in $\tilde{H}_{k-1}(K)$ must be supported on at least $2k$ vertices and, if the support size is exactly $2k$, then the class must be given by an embedded O_k , see [14]. This leads to our interest in their algorithmic properties.

There are two equivalent view-points of flag complexes. Geometers prefer to think of them as clique complexes of graphs that encode some incidence relation, e.g., proximity in a metric space (the Vietoris-Rips complex). We will follow the combinatorial topologists' approach via independence complexes of graphs. For a graph G , the *independence complex* $I(G)$ is the simplicial complex whose vertices are the vertices of G and whose faces are the independent sets of G . Note that O_k is the independence complex of the disjoint union of k edges. This immediately leads to a characterization of cross-cycles in $I(G)$ in terms familiar to graph theorists.

Observation. If G is a graph then a cross-cycle of size k in $I(G)$ determines, and is determined, by an induced matching M of size k in G such that the vertex set of M contains a subset D that is a maximal independent set in G . For simplicity we will say 'an induced matching containing a maximal independent set'.

Note that D must contain exactly one vertex from each edge of M and that D is a dominating set of G (see Section 2). For example, in the graph G of Fig. 1, $M = \{(7, 9), (3, 8)\}$ is an induced matching containing an independent set $D = \{7, 3\}$ which is maximal in G . That defines a cross-cycle and therefore a nontrivial homology class in $\tilde{H}_1(I(G))$.

For our algorithmic analysis we always assume that a flag complex is presented as $I(G)$ and the graph G is the input. Our first observation is that finding a cross-cycle is hard in general.

Theorem 1. *Given a graph G , it is NP-complete to decide if G has an induced matching containing a maximal independent set. Consequently, it is NP-complete to decide if $I(G)$ has a cross-cycle.*

We then concentrate on chordal graphs. Quite a lot is known about their independence complexes. In particular, they are vertex-decomposable [23] and therefore homotopy equivalent to wedges of spheres (also proved in [22, 16]). Moreover, every wedge of spheres arises, up to homotopy, as an independence complex of a chordal graph [16]. Another reason to study chordal graphs in this context is that for this family of graphs cross-cycles detect all of the homology of the independence complex.

Theorem 2. *For a chordal graph G and any $k \geq 0$, the homology group $\tilde{H}_{k-1}(I(G))$ is non-trivial if and only if G has an induced matching of size k containing a maximal independent set.*

Since a wedge of spheres is completely determined by the ranks of homology groups, Theorem 2 implies that for a chordal graph G the whole homotopy type of $I(G)$ is encoded in the cross-cycles. The next two results describe quite precisely how much of that information can be recovered from G in polynomial time.

Theorem 3. *There is a polynomial time algorithm that decides, for a chordal graph G , if G has an induced matching containing a maximal independent set.*

Theorem 4. *Given a chordal graph G and an integer k , it is NP-complete to decide if G has an induced matching of size k containing a maximal independent set.*

By Theorem 2 these results have immediate topological corollaries.

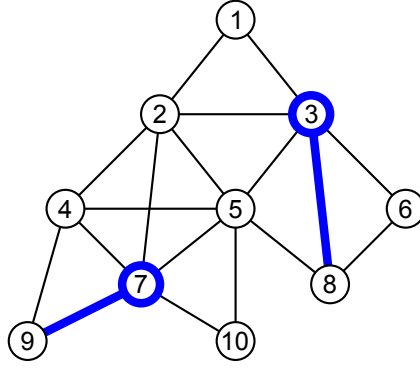


Figure 1: A sample graph with an induced matching containing a maximal independent set.

Theorem 5. *For a chordal graph G one can decide in polynomial time*

- a) *if $I(G)$ is contractible,*
- b) *if $I(G)$ is simply-connected.*

Theorem 6. *Given a chordal graph G and integer k , it is NP-complete to decide if the group $\tilde{H}_{k-1}(I(G))$ is non-trivial.*

Let us briefly comment on these results. Theorem 6 implies that in general computing the homology groups of flag complexes is NP-hard with the one-skeleton as input. That this must be the case is not surprising, as such a calculation requires the enumeration of (exponentially many) k -faces, and there does not seem to be a significantly faster method, although there is much research on practical algorithms and their performance (see [26] and the references therein). However, we are not aware of any previous proof that this problem is in fact hard.

Also the positive results break some barriers. First, note that for an arbitrary graph G the problems of deciding if $I(G)$ is simply-connected or contractible are both undecidable [6]. Moreover all previous work on topological features of chordal graphs [23, 5, 16, 22] made use exclusively of the existence of simplicial vertices (see Section 2). Our algorithm in Theorem 5 makes essential use of the geometric intersection representation of chordal graphs and their tree models. Finally, we remark that a straightforward application of Lemma 10 yields an algorithm that solves the problem of Theorem 3 in linear time for forests. This improves the result of [17] and answers a question of [7]. If G is a forest we obtain a complete information about the homotopy type of $I(G)$, since it is known to be homotopy equivalent to a single sphere or contractible [7].

Finally, one can ask for natural subclasses of chordal graphs for which also the dimensions of non-trivial homology groups could be recovered in polynomial time. We prove such results for interval graphs and chordal graphs of bounded leafage (see Section 6 for the definitions).

Theorem 7. *There is a polynomial time algorithm that decides, for an interval graph G and integer k , if G has an induced matching of size k containing a maximal independent set.*

Theorem 8. *There is a $O(k^\ell n^{\ell+2})$ time algorithm that decides, for an chordal graph G of leafage ℓ and integer k , if G has an induced matching of size k containing a maximal independent set.*

The rest of the paper is set out as follows. In Sections 2 and 3, we provide the necessary prerequisites in graph theory and combinatorial topology. In Section 4, we prove Theorem 2 and use it to deduce Theorems 5, 6 from the corresponding combinatorial results. Our main algorithm from Theorem 3 is presented in Section 5 and its extensions (Theorems 7, 8) are described in Section 6. Finally in Section 7, we prove the hardness results of Theorems 1, 4.

2 Preliminaries on graph theory

A graph $G = (V, E)$ with vertex set $V(G)$ and edge set $E(G) \subseteq V(G) \times V(G)$ is always finite, undirected, with no loops or parallel edges. The *neighbourhood* of a vertex $v \in V(G)$ is defined as $N(v) = \{u \mid uv \in E(G)\}$. We write $N[v]$ for $N(v) \cup \{v\}$. For a set $X \subseteq V(G)$, we define $N[X] = \bigcup_{v \in X} N[v]$ and write $N(X)$ for $N[X] \setminus X$. We denote by $G[X]$ the subgraph of G induced on X , i.e. the graph $G[X] = (X, E \cap X \times X)$. Further, we write $G - X$ for the graph $G[V(G) \setminus X]$.

A set $X \subseteq V(G)$ is *dominating* if $N[X] = V(G)$. A *clique* of G is a set of pairwise adjacent vertices of G . An *independent set* of G is a set of pairwise non-adjacent vertices of G . A vertex $v \in V(G)$ is *simplicial* if $N(v)$ is a clique of G . A set of edges of G is a *matching* if no two edges in the set share a common endpoint. A matching M of G is an *induced matching* if the subgraph of G induced by the endpoints of the edges in M contains only the edges in M . Simply put a matching of G is induced if it occurs as an induced subgraph of G .

A graph is *chordal* if it contains no induced cycle of length four or longer. By [4], every chordal graph has a simplicial vertex. Removing it yields a smaller chordal graph, which gives rise to the so-called *perfect elimination ordering*. Such orderings only exist for chordal graphs [11].

A *tree model* of a graph $G = (V, E)$ consists of a tree T , called a *host tree*, and a collection of subtrees of T associated with the vertices of G , denoted by $\{T_u\}_{u \in V}$, with the property that $uv \in E$ if and only if $V(T_u) \cap V(T_v) \neq \emptyset$. In the interest of clarity, we shall use capital letters X, Y, \dots for the vertices of T and call them *nodes*. The following is a well-known fact [10].

Theorem 9. [10] *A graph is chordal if and only if it has a tree model.*

We remark that there exists a linear time algorithm [24, 11] to determine whether an input graph G is chordal, and if so, to construct a tree model of G .

For further notation and terminology we invite the reader to consult [11, 25].

3 Preliminaries on combinatorial algebraic topology

We now recall some basic notions of algebraic topology and homology theory. An excellent concise introduction is [21, Chapter 0.3]. More details can be found in [18].

A *simplicial complex* K with vertex set V is a collection of subsets of V such that if $B \in K$ and $A \subseteq B$ then $A \in K$. The elements of K are called *faces*. For a face $\sigma \in K$ we denote by $|\sigma|$ its cardinality (number of vertices) and by $\dim \sigma = |\sigma| - 1$ its dimension. We think of K as of the triangulation of a topological space (namely the geometric realization of K). The *join* of two simplicial complexes K and L with disjoint vertex sets is the complex $K * L = \{\sigma \cup \tau : \sigma \in K, \tau \in L\}$. It is also convenient to treat the empty simplicial complex (with no vertices and the unique face \emptyset) as the (-1) -dimensional sphere S^{-1} .

A simplicial complex K is called *flag* if every minimal non-face of K has dimension 1. This is equivalent to saying that K is the clique complex of its 1-dimensional skeleton or the independence complex of the complement of that skeleton.

Two topological spaces X, Y are *homotopy equivalent* or *have the same homotopy type*, which we denote $X \simeq Y$ if there are continuous maps $f : X \rightarrow Y$ and $g : Y \rightarrow X$ such that $fg \sim \text{id}_Y$ and $gf \sim \text{id}_X$ where \sim denotes the relation of *homotopy* between maps [18]. Loosely speaking it means that one can continuously deform one map to the other. This is a relaxation of the notion of homeomorphism. A space is *contractible* if it is homotopy equivalent to a point.

We now proceed to define simplicial homology. An *oriented face* of K is a face σ together with an equivalence class of orderings of the vertices of σ , where two orderings are equivalent if they differ by an even permutation. We will denote by $[v_0, \dots, v_p]$ the oriented face corresponding to the face $\sigma = \{v_0, \dots, v_p\}$ together with the ordering $v_0 < \dots < v_p$. Let $C_p(K)$ be

the free abelian group with basis consisting of the oriented faces of dimension p modulo the relations $\sigma_1 + \sigma_2 = 0$ whenever σ_1 and σ_2 are oriented faces corresponding to the same face σ but with opposite orientations. The elements of $C_p(K)$ are called p -chains and $C_p(K) = 0$ when $p \leq -2$. There are maps $\partial_p : C_p(K) \rightarrow C_{p-1}(K)$ defined on the basis elements via

$$\partial_p[v_0, \dots, v_p] = \sum_{i=0}^p (-1)^i [v_0, \dots, v_{i-1}, v_{i+1}, \dots, v_p].$$

Then ∂_p extend to group homomorphisms and $\partial_p \partial_{p+1} = 0$. The elements of $\ker \partial_p \subseteq C_p(K)$ are called *cycles* and the elements of $\text{im } \partial_{p+1} \subseteq C_p(K)$ are *boundaries*. The p -th reduced homology group of K is the quotient $\tilde{H}_p(K) = \ker \partial_p / \text{im } \partial_{p+1}$. If K is finite, it is a finitely generated abelian group whose rank is called the p -th reduced Betti number $\tilde{b}_p(K)$. The homology groups are an invariant of the homotopy type of the space.

We can now describe the homology classes defined by cross-cycles in flag complexes. First of all, suppose $M = \{(v_{1,1}, v_{1,2}), (v_{2,1}, v_{2,2}), \dots, (v_{k,1}, v_{k,2})\}$ is an induced matching in a graph G . Then we define a $(k-1)$ -chain $\alpha_M \in C_{k-1}(I(G))$ by

$$\alpha_M = \sum_{\epsilon_1, \dots, \epsilon_k \in \{1,2\}} (-1)^{\epsilon_1 + \dots + \epsilon_k} [v_{1,\epsilon_1}, v_{2,\epsilon_2}, \dots, v_{k,\epsilon_k}].$$

Since M is an induced matching, each collection $\{v_{1,\epsilon_1}, \dots, v_{k,\epsilon_k}\}$ is an independent set in G , so the definition makes sense. We leave it to the reader to verify that $\partial_{k-1} \alpha_M = 0$, therefore α_M is a cycle and it determines a homology class $[\alpha_M] \in \tilde{H}_{k-1}(I(G))$. Now assume that M contains a maximal independent set D of G , which we can without loss of generality assume to be $D = \{v_{1,1}, v_{2,1}, \dots, v_{k,1}\}$. Then the face $[v_{1,1}, v_{2,1}, \dots, v_{k,1}]$ is maximal in $I(G)$, so it does not appear in the image of $\partial_k : C_k(I(G)) \rightarrow C_{k-1}(I(G))$ and it follows that the homology class $[\alpha_M] \in \tilde{H}_{k-1}(I(G))$ represented by α_M is non-zero.

Unfortunately it is not the case that $\tilde{H}_{k-1}(I(G))$ is generated by the classes $[\alpha_M]$ for induced matchings M of size k . For example it is easy to see that $I(C_5) = S^1$, but C_5 does not even have induced matchings of size two.

In the next section, we show that the situation is much better for chordal graphs.

4 Homology for chordal graphs

In this section we prove that if G is chordal then all of the homology of $I(G)$ can be detected using cross-cycles.

Recall that for a sequence of topological spaces X_1, \dots, X_n the *wedge sum* (or one-point union) $\bigvee_{i=1}^n X_i$ is the space obtained from the disjoint union of the X_i 's by choosing a sequence of distinguished points $x_i \in X_i$ and identifying them to a single point. If $n = 0$, we declare the wedge sum of an empty family to be the space consisting of one point. For a simplicial complex K , the *suspension* is $\Sigma K = S^0 * K$. These constructions are well-behaved with respect to reduced homology: we have $\tilde{H}_k(\bigvee_{i=1}^n X_i) = \bigoplus_{i=1}^n \tilde{H}_k(X_i)$ and $\tilde{H}_k(\Sigma K) = \tilde{H}_{k-1}(K)$. In fact these properties are usually taken as the axioms of a reduced homology theory.

We are going to use the following result of Engström [5].

Lemma 10. [5] *If G is any graph and v is a simplicial vertex then there is a homotopy equivalence*

$$I(G) \simeq \bigvee_{u \in N(v)} \Sigma I(G \setminus N[u]).$$

Theorem 11 (Theorem 2). *For a chordal graph G and any $k \geq 0$, we have that $\tilde{H}_{k-1}(I(G)) \neq 0$ if and only if G has an induced matching of size k containing a maximal independent set.*

Proof. The ‘if’ part follows from the discussion of cross-cycles in the previous section. To prove the ‘only if’ part we use induction on the number of vertices in G .

If $V(G) = \emptyset$, then $I(G) = S^{-1}$ and $\tilde{H}_{-1}(S^{-1}) = \mathbb{Z}$. In this case the empty matching of size 0 satisfies all the requirements. If G has an isolated vertex, then $I(G)$ is contractible and $\tilde{H}_{k-1}(I(G)) = 0$, so there is nothing to do. This completes the check of initial conditions.

Otherwise suppose that G is a chordal graph with no isolated vertices and $\tilde{H}_{k-1}(I(G)) \neq 0$. Let v be any simplicial vertex of G (which exists by [4]). By Lemma 10 there is a splitting

$$\tilde{H}_{k-1}(I(G)) = \bigoplus_{u \in N(v)} \tilde{H}_{k-2}(I(G \setminus N[u])).$$

It follows that there exists a vertex $u \in N(v)$ such that $\tilde{H}_{k-2}(I(G \setminus N[u])) \neq 0$. The graph $G \setminus N[u]$ is chordal, so by induction it has an induced matching M' of size $k-1$ containing a maximal independent set D' . Now define a new pair (M, D) of size k in G by setting

$$M = M' \cup \{vu\}, \quad D = D' \cup \{u\}.$$

We easily see that M is an induced matching in G and D is an independent set which is maximal in G . This completes the proof. \square

Remark. A more careful analysis of this argument shows that, in fact, slightly more is true. We leave the proof to the reader, as we do not need the full strength of the next result for our algorithmic applications.

Proposition 12. *If G is chordal and $k \geq 0$ then the homology group $\tilde{H}_{k-1}(I(G))$ is generated by the classes $[\alpha_M]$ as M runs through all induced matchings of size k containing maximal independent sets.*

We close this section by proving Theorems 5 and 6.

Proof of Theorems 5 and 6. Since for a chordal graph G the space $I(G)$ has the homotopy type of a wedge sum of spheres [23], it is contractible if and only if all its homology groups vanish. Therefore, the first part of Theorem 5 follows directly from Theorem 3 and 2. The second part is a consequence of the fact that simply-connectedness of $I(G)$ is equivalent to the vanishing of $\tilde{H}_1(I(G))$, which by Theorem 2 is equivalent to G not having an induced matching of size two containing a maximal independent set. Clearly this condition can be checked in polynomial time. Theorem 6 follows from Theorems 4 and 2. \square

5 Algorithm for chordal graphs

In this section, we prove Theorem 3, that is, we present a polynomial time algorithm checking if a chordal graph G has an induced matching M containing a maximal independent set D . For simplicity, we call the pair (M, D) a *solution* to G .

Let G be a chordal graph. Recall that one can construct a tree model of G in linear time [24]. Thus, for the rest of this section, we shall assume that we have a fixed tree model of G and all subsequent considerations are always with respect to this model.

Namely, let a tree T with a collection $\{T_u\}_{u \in V(G)}$ of subtrees of T be a tree model of G . We consider T rooted at some node, and we direct all edges of T away from the root (Fig. 2). This allows us to treat T as an upper semi-lattice. In particular, for $X, Y \in V(T)$, we write $X \preceq Y$ if there is in T a directed path from Y to X . Observe that \preceq is a partial order. We write $X \prec Y$ if $X \preceq Y$ and $X \neq Y$. If XY is an edge of T oriented from Y to X , we say that Y is the *parent* of X , and X is a *child* of Y . If $X \prec Y$, we say that Y is an *ancestor* of X , and X is a *descendant* of Y .

Notation. For $u \in V(G)$, $\text{top}(u)$ denotes the maximum element of $V(T_u)$ with respect to \preceq .

The following is a simple consequence of the definition of a tree model (proof in appendix).

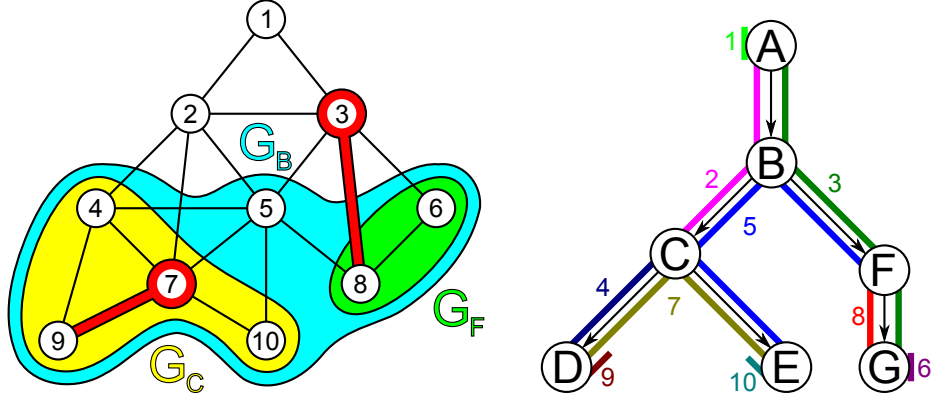


Figure 2: A sample chordal graph G and its rooted tree model.

Lemma 13. *If $uv \in E(G)$, then $\text{top}(u) \succeq \text{top}(v) \in V(T_u)$ or $\text{top}(v) \succeq \text{top}(u) \in V(T_v)$.*

To decide whether a solution to G exists, it suffices to consider particular type of solutions.

Definition. A solution (M, D) to G is *canonical* if every edge uv in M is such that $u \in D$ if $\text{top}(u) \succ \text{top}(v)$, and $v \in D$ if $\text{top}(u) \prec \text{top}(v)$.

The following is an important result which is ultimately a consequence of Lemma 10. (The proof can be found in the appendix.)

Lemma 14. *If there exists a solution to G , then there exist a canonical solution to G of the same size.*

Our algorithm is based on dynamic programming on T that tries to find canonical solutions for subgraphs of G and then combines these solutions to obtain a solution to G if one exists. In particular, we focus on subgraphs induced by vertices whose subtrees lie completely below some node of T (see Fig. 2).

Notation. For $X \in V(T)$, G_X denotes the subgraph of G induced on vertices v with $\text{top}(v) \preceq X$. If X is the root of T , then $G_X = G$. We distinguish the following special type of solutions to G_X .

Definition. A solution (M, D) to G_X is *rooted* if there exists $u \in D$ such that $\text{top}(u) = X$.

Notation. \mathcal{S} denotes the set of nodes $X \in V(T)$ such that there exists a solution to G_X and \mathcal{R} denotes the set of nodes $X \in V(T)$ such that there exists a rooted canonical solution to G_X .

Let us explain the notation and our strategy using the example in Fig. 2. We want to know which nodes are in \mathcal{S} (namely, whether the root of T is in \mathcal{S}). To do so, we recursively find the nodes that admit a rooted solution (the set \mathcal{R}). For example, $C \in \mathcal{R}$ since the subgraph G_C has a solution $(\{(7, 9)\}, \{7\})$. On the other hand, $B \notin \mathcal{R}$ since for any choice of an edge $(5, v)$ in G_B , the graph $G_B \setminus (N[5] \cup N[v])$ has an isolated vertex, and thus no solution. Still, G_B has a non-rooted solution combined from the rooted solutions of the disjoint subgraphs G_C and G_F . Finally for $G = G_A$, we try the edge $(3, 8)$ and see that $G_A \setminus (N[3] \cup N[8]) = G_C$. We then check that the rooted solution to G_C together with the edge $(3, 8)$ indeed form a solution to G_A .

In the rest of this section, we present three technical lemmas that explain how we can compute the sets \mathcal{R} and \mathcal{S} . Namely, Lemma 16 explains how to obtain a solution to G_X (if one exists) by combining rooted canonical solutions for descendants of X , and Lemma 17 provides a way to determine whether a rooted solution exists to G_X (for proofs see the appendix).

After the lemmas, we finally present our algorithm and analyze its complexity.

Lemma 15. Let $X \in V(T)$ and suppose that there exists a canonical solution (M, D) to G_X . Then

- (i) if $Y = \text{top}(u)$ for some $u \in D$, then $Y \in \mathcal{R}$, and
- (ii) if $Y \preceq X$ and $Y \notin \bigcup_{u \in D} V(T_u)$, then $Y \in \mathcal{S}$.

Lemma 16. Let $X \in V(T)$. Define $\mathcal{R}_X = \{Y \mid Y \preceq X, Y \in \mathcal{R}\}$ and let \mathcal{R}_X^* denote the set of maximal elements of \mathcal{R}_X with respect to \preceq . For each $Y \in \mathcal{R}_X^*$, let (M_Y, D_Y) be a rooted canonical solution to G_Y , and define

$$M^* = \bigcup_{Y \in \mathcal{R}_X^*} M_Y \quad D^* = \bigcup_{Y \in \mathcal{R}_X^*} D_Y$$

Then, if there exists a solution to G_X , then (M^*, D^*) is a canonical solution to G_X .

Lemma 17. Let $Z \in V(T)$, and let $uv \in E(G)$ be an edge such that $\text{top}(u) = Z$ and $\text{top}(v) \preceq Z$. Define $\mathcal{L}_{uv} = \{Y \mid Y \preceq Z, Y \notin V(T_u) \cup V(T_v)\}$ and let \mathcal{L}_{uv}^* be the set of maximal elements of \mathcal{L}_{uv} with respect to \preceq . For each $X \in \mathcal{L}_{uv}^*$, define \mathcal{R}_X^* just like in Lemma 16, and for each $Y \in \mathcal{R}_X^*$, let (M_Y, D_Y) be a rooted canonical solution to G_Y . Finally, define

$$M_{uv}^* = \{uv\} \cup \bigcup_{X \in \mathcal{L}_{uv}^*} \bigcup_{Y \in \mathcal{R}_X^*} M_Y \quad D_{uv}^* = \{u\} \cup \bigcup_{X \in \mathcal{L}_{uv}^*} \bigcup_{Y \in \mathcal{R}_X^*} D_Y$$

Then, if there exists a (rooted) canonical solution (M, D) to G_Z such that $uv \in M$ and $u \in D$, then (M_{uv}^*, D_{uv}^*) is a rooted canonical solution to G_Z .

5.1 Proof of Theorem 3

The algorithm processes all nodes of T from leaves to the root as follows.

For each $Z \in V(T)$ such that all descendants of Z have already been processed, we try to decide whether there exists a rooted canonical solution to G_Z . To do this, we apply Lemma 17. Namely, we try every edge $uv \in E(G)$ such that $\text{top}(u) = Z$ and $\text{top}(v) \preceq Z$. We construct the sets M_{uv}^* and D_{uv}^* as described in Lemma 17. Note that we are able to construct these sets because all descendants of Z have already been processed. Then we test if D_{uv}^* is a dominating set of G_Z . If so, we declare that there exists a rooted canonical solution to G_Z . If we fail for every possible choice of uv , then we declare that there is no rooted canonical solution to G_Z . The correctness of this procedure is guaranteed by Lemma 17.

After processing all nodes of T , we apply Lemma 16 for the root. Namely, we construct the sets M^* and D^* as described in the lemma (for $X = \text{root of } T$) and test if D^* is a dominating set of G . If so, we declare that G has a solution and we return the pair (M^*, D^*) . If not, we declare that no solution for G exists. The correctness of this step is guaranteed by Lemma 16.

The algorithm can be implemented to run in time $O(|E(G)|^2)$ (details in the appendix.) \square

6 Extensions and small cases

In this section, we briefly examine a modification of our algorithm from Section 5 which will allow us to find exact-size solutions in restricted classes. This will imply Theorems 7 and 8.

We shall use the notation from Section 5. Again, we have a chordal graph G and a tree model for G consisting of a tree T and subtrees $\{T_u\}_{u \in V(G)}$. Similarly, we define the partial order \preceq on $V(T)$, and the subgraphs G_X .

The following two technical lemmas are analogues of Lemmas 16 and 17 for deciding if there exists a solution of size k . Their proofs are straightforward, and similar to the proofs of Lemmas 16 and 17 so we omit the details. After the lemmas, we discuss their consequences.

Notation. $\mathcal{R}^{(k)}$ denotes the set of nodes $X \in V(T)$ such that there exists a rooted canonical solution to G_X of size k , and $\mathcal{R}_X^{(k)}$ for $X \in V(T)$ denotes the set $\{Y \mid Y \preceq X, Y \in \mathcal{R}^{(k)}\}$.

Lemma 18. Let $X \in V(T)$. Suppose that there is a solution to G_X of size k . Then there exist positive integers k_1, k_2, \dots, k_t and nodes Y_1, \dots, Y_t where $Y_i \in \mathcal{R}_X^{(k_i)}$ for all $i \in \{1, \dots, t\}$ such that

- (i) $k_1 + k_2 + \dots + k_t = k$,
- (ii) Y_1, Y_2, \dots, Y_t are pairwise incomparable with respect to \preceq , and
- (iii) if (M_{Y_i}, D_{Y_i}) is any rooted canonical solution to G_{Y_i} of size k_i for each $i \in \{1, \dots, t\}$, and

$$M^* = \bigcup_{i=1}^t M_{Y_i} \quad D^* = \bigcup_{i=1}^t D_{Y_i}$$

then (M^*, D^*) is a canonical solution to G_X of size k .

Notation. For $uv \in E(G)$ with $\text{top}(u) \succeq \text{top}(v)$, we define $\mathcal{R}_{uv}^{(k)} = \mathcal{R}_{\text{top}(u)}^{(k)} \setminus (V(T_u) \cup V(T_v))$.

Lemma 19. Let $X \in V(T)$, and let $uv \in E(G)$ be an edge such that $\text{top}(u) = X$ and $\text{top}(v) \preceq X$. Suppose that there is a canonical solution (M, D) to G_X such that $uv \in M$ and $u \in D$. Then there exist positive integers k_1, k_2, \dots, k_t and nodes Y_1, \dots, Y_t where $Y_i \in \mathcal{R}_{uv}^{(k_i)}$ for all $i \in \{1, \dots, t\}$ such that

- (i) $k_1 + k_2 + \dots + k_t = k - 1$,
- (ii) Y_1, Y_2, \dots, Y_t are pairwise incomparable with respect to \preceq , and
- (iii) if (M_{Y_i}, D_{Y_i}) is any rooted canonical solution to G_{Y_i} of size k_i for each $i \in \{1, \dots, t\}$, and

$$M^* = \{uv\} \cup \bigcup_{i=1}^t M_{Y_i} \quad D^* = \{u\} \cup \bigcup_{i=1}^t D_{Y_i}$$

then (M^*, D^*) is a rooted canonical solution to G_X of size k .

We now look at how these lemmas can be applied in special subclasses of chordal graphs.

First, we look at *interval* graphs. These are defined as the intersection graphs of intervals of the real line. Every interval graph G is chordal and, in particular, the intersection model of intervals allows us to choose a tree model of G such that \preceq is a total order. In particular, this means that the largest set of pairwise incomparable nodes of T with respect to \preceq is of size one. It follows that we can compute the sets $\mathcal{R}^{(k)}$ in polynomial time as it suffices to apply Lemmas 18 and 19 with $t \in \{0, 1\}$. This proves Theorem 7 (for complexity analysis see below).

More generally, we consider chordal graphs of bounded *leafage*. The leafage of a chordal graph G is defined as the smallest integer ℓ such that G has a tree model with ℓ leaves. It was first defined in [19] and it can be computed in polynomial time by the algorithm of [12]. The same algorithm also constructs a tree model of G with minimum number of leaves.

Thus we may assume that T has exactly ℓ leaves. Similarly to the case of interval graphs, this implies that the largest set of pairwise incomparable nodes of T with respect to \preceq has size ℓ . Thus to compute the sets $\mathcal{R}^{(k)}$ it suffices to apply Lemmas 18 and 19 for all $t \in \{0, 1, \dots, \ell\}$ which again yields a polynomial time algorithm. Namely, the complexity is $O(k^\ell n^{\ell+2})$ which is $O(kn^3)$ for the case of interval graphs (details in the appendix.) This proves Theorem 8.

7 Hardness

In this section, we sketch the constructions used in the proofs of hardness results. Namely, we prove Theorems 1 and 4 as outlined in the introduction.

7.1 Proof of Theorem 1 (sketch)

The problem is clearly in NP. To prove that it is also NP-hard we construct a reduction from 3SAT which is NP-hard [9]. Consider an instance \mathcal{I} to 3SAT, namely clauses C_1, \dots, C_m using variables v_1, \dots, v_n . By standard arguments, we may assume that no variable appears twice in the same clause (positively or negatively). We write \bar{v}_i for the negation of v_i . From \mathcal{I} we construct a graph $G_{\mathcal{I}}$ as follows. The vertex set of $G_{\mathcal{I}}$ is $\{c_1, \dots, c_m, v_1, \dots, v_n, \bar{v}_1, \dots, \bar{v}_n\}$. It contains edges $v_1\bar{v}_1, \dots, v_n\bar{v}_n$ and there is an edge between v_i (resp. \bar{v}_i) and c_j if the literal v_i (resp. \bar{v}_i) appears in the clause C_j . Finally, the set $\{c_1, \dots, c_m\}$ forms a clique.

We prove that \mathcal{I} has a satisfying truth assignment if and only if $G_{\mathcal{I}}$ has an induced matching containing a maximal independent set. If $\varphi : \{v_1, \dots, v_n\} \rightarrow \{\text{true}, \text{false}\}$ is a satisfying truth assignment, we define $M = \{v_1\bar{v}_1, \dots, v_n\bar{v}_n\}$ and $D = \{v_i \mid \varphi(v_i) = \text{true}\} \cup \{\bar{v}_i \mid \varphi(v_i) = \text{false}\}$. By construction, M is an induced matching containing D , and D is an independent set and is maximal because every clause vertex c_j is adjacent to a literal that evaluates to true. Conversely, if $G_{\mathcal{I}}$ has an induced matching M containing a maximal independent set D , we first observe that $D \cap \{c_1, \dots, c_m\} = \emptyset$ and exactly one of v_i, \bar{v}_i belongs to D for each i . We assign true to all literals that are in D , and then observe that this constitutes a satisfying truth assignment. \square

7.2 Proof of Theorem 4

To prove Theorem 4, we first prove a somewhat stronger result concerning independent dominating sets in chordal graphs. Note that independent dominating sets are precisely maximal independent sets. Finding a smallest or largest such set is possible in polynomial time in chordal graphs [8, 11]. If vertices have weights, then finding such a set of maximum weight is still polynomial, but for minimum weight the problem is already NP-hard [2]. We prove that finding such a set of exact cardinality is also NP-hard. This will imply Theorem 4.

Lemma 20. *It is NP-hard to decide, for a given chordal graph G and integer k , whether or not G has a maximal independent set of size k .*

Proof (sketch). The proof is by reduction from the exact cardinality dominating set problem in general graphs which is NP-hard, since minimum size dominating set problem is NP-hard [9].

Namely, consider a graph $G = (V, E)$ and an integer k . Construct the following graph G' .

$$V(G') = V \cup \{v_1, v_2, v_3, v_4, v_5, v_6 \mid v \in V\}$$

$$E(G') = \{uv_1 \mid uv \in E\} \cup \{u_1v_1 \mid u, v \in V\} \cup \{vv_1, vv_2, vv_4, vv_5, vv_6, v_2v_3 \mid v \in V\}$$

Observe that G' is a chordal graph. Further, G has a dominating set of size k if and only if G' has an independent dominating set of size $4|V| - 2k$. That concludes the proof. \square

Now, we are ready to prove Theorem 4 by reduction from the above problem. Consider a chordal graph $G = (V, E)$ and an integer k . We construct a graph G' from G by substituting a clique of size two for every vertex of G . The graph G' is defined as follows.

$$V(G') = V \cup \{v_1 \mid v \in V\}$$

$$E(G') = \{vv_1 \mid v \in V\} \cup \{uv, uv_1, u_1v, u_1v_1 \mid uv \in E\}$$

Observe that G' is a chordal graph, since chordal graphs are closed under clique substitution. Moreover, G has an independent dominating set of size k if and only if G' has an induced matching of size k containing a maximal independent set. That proves Theorem 4. \square

Remark. Note that our hardness results do not address the problem of finding *minimum* or *maximum* size cross-cycles in the case of chordal graphs. By Theorem 2 finding the minimum size cross-cycle is equivalent to calculating the *connectivity* of $I(G)$ for a chordal graph G . We believe that these two problems are also hard, but we were not able to come up with a proof.

References

- [1] M. Adamaszek, *Special cycles in independence complexes and superfrustration in some lattices*, arxiv/1109.4914
- [2] Gerard J. Chang, *The weighted independent domination problem is NP-complete for chordal graphs* Discrete Applied Mathematics 143 (2004) 351-352
- [3] P. Csorba, *Subdivision yields Alexander duality on independence complexes*, Electr. J. Comb. 16(2) (2009), #R11
- [4] G. A. Dirac, *On rigid circuit graphs*, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg, 25 (1961) 71-76
- [5] A. Engström, *Complexes of directed trees and independence complexes*, Discrete Math. 309 (2009), 3299-3309
- [6] W. Haken, *Connections between topological and group theoretical decision problems*, In: Boone, Cannonito and Lyndon (1973), 427-441.
- [7] R. Ehrenborg, G. Hetyei. *The topology of the independence complex*, Eur. J. Comb. 27(6): 906-923 (2006)
- [8] M. Farber, *Independent domination in chordal graphs*, Operations Research Letters 1 (1982) 134-138
- [9] M. R. Garey, D. S. Johnson, **Computers and Intractability: A Guide to the Theory of NP-Completeness**, W. H. Freeman, 1979
- [10] F. Gavril, *The intersection graphs of subtrees in trees are exactly the chordal graphs*, Journal of Combinatorial Theory B 16 (1974), 47-56
- [11] M. C. Golumbic, **Algorithmic Graph Theory and Perfect Graphs (2nd edition)**, North Holland, 2nd ed., 2004
- [12] M. Habib, J. Stacho, *Polynomial-time algorithm for the leafage of chordal graphs*, Algorithms - ESA 2009, Lecture Notes In Computer Science 5757 (2009) 290-300
- [13] J. Jonsson, *Certain Homology Cycles of the Independence Complex of Grids*, Discrete & Computational Geometry 43(4): 927-950 (2010)
- [14] M. Kahle, *Topology of random clique complexes*, Discrete Math. 309(6): 1658-1671 (2009)
- [15] M. Kahle, E. Meckes, *Limit theorems for Betti numbers of random simplicial complexes*, preprint arxiv/1009.4130
- [16] K. Kawamura. *Independence complexes of chordal graphs* Discrete Math. 310 (2010), 2204-2211
- [17] K. Kawamura. *Homotopy types of independence complexes of forests*, Contr. to Disc. Math. 5 (2010), 67-75
- [18] D. Kozlov, *Combinatorial Algebraic Topology*, Algorithms and Computation in Mathematics, Vol. 21, Springer-Verlag Berlin Heidelberg 2008
- [19] I.-J. Lin, T. A. McKee, D. B. West: *The leafage of a chordal graph*, Discussiones Mathematicae Graph Theory 18 (1998) 23-48

- [20] J. Shareshian, M. L. Wachs, *Torsion in the matching complex and chessboard complex*, Adv. Math. 212 (2007), no. 2, 525-570.
- [21] R. P. Stanley, *Combinatorics and commutative algebra*, Progress in Mathematics 41, Birkhauser, 1983
- [22] A. Van Tuyl, R. H. Villarreal, *Shellable graphs and sequentially Cohen-Macaulay bipartite graphs*, J. Combin. Theory Ser. A 115 (2008), no. 5, 799-814
- [23] R. Woodroffe, *Vertex decomposable graphs and obstructions to shellability*, Proc. Amer. Math. Soc. 137 (2009), No. 10, 3235-3246
- [24] D. J. Rose, R. E. Tarjan, G. S. Leuker, *Algorithmic aspects of vertex elimination on graphs*, SIAM Journal on Computing 5 (1976) 266-283
- [25] D. B. West, **Introduction to Graph Theory**, Prentice Hall, 2nd ed., 2000.
- [26] A. Zomorodian, *The tidy set: a minimal simplicial set for computing homology of clique complexes*, Symposium on Computational Geometry 2010, 257-266

A Appendix

Before proceeding with the omitted proofs, we mention an important observation regarding simplicial vertices which we make use of later in the proofs.

Lemma 21. *Let G be a graph and v be a simplicial vertex of G . Suppose that G has an induced matching M containing a maximal independent set D .*

- (i) *If $v \in D$, then $uv \in M$ for some $u \in N(v)$ and we define $M' = M$ and $D' = D \setminus \{v\} \cup \{u\}$.*
- (ii) *If $v \notin D$, then $u \in D$ and $uw \in M$ for some $u \in N(v)$ and $w \in N(u)$ (possibly $w = v$), and we define $M' = M \setminus \{uw\} \cup \{uv\}$ and $D' = D$.*

Then M' is an induced matching of G , D' is a maximal independent set of G and M' contains D' .

A.1 Omitted Proofs

Proof of Lemma 13. Since $uv \in E(G)$, we conclude that $V(T_u) \cap V(T_v)$ is a non-empty set. Let Y be any maximal element of $V(T_u) \cap V(T_v)$ with respect to \preceq . Since $\text{top}(u)$ is the unique maximal element of $V(T_u)$, we conclude $Y \preceq \text{top}(u)$. Similarly, $Y \preceq \text{top}(v)$.

Suppose that $Y \notin \{\text{top}(u), \text{top}(v)\}$. Then Y cannot be the root of T , and hence, it has a parent Z in T . Recall that T_u is a connected subgraph of T . Thus, every node on the path from Y to $\text{top}(u)$ belongs to $V(T_u)$. In particular, Z belongs to this path and thus belongs to $V(T_u)$. Similarly, Z belongs to $V(T_v)$. So $Z \in V(T_u) \cap V(T_v)$ which contradicts the maximality of Y .

Hence, we conclude that $Y = \text{top}(u)$ or $Y = \text{top}(v)$. But now this exactly yields the claim. (To see this recall that $Y \in V(T_u) \cap V(T_v)$ and that $Y \preceq \text{top}(u)$ and $Y \preceq \text{top}(v)$.) \square

Proof of Lemma 14. The proof is by induction on G . Suppose that there exists a solution (M, D) to G . Let v be a vertex of G such that $\text{top}(v)$ is a minimal element of the set $\{\text{top}(u) \mid u \in V(G)\}$ with respect to \preceq . We claim that v is a simplicial vertex of G . Suppose otherwise, namely that there are non-adjacent vertices $u, w \in N(v)$. By Lemma 13 and the minimality of v , we conclude that $\text{top}(v) \in T_u$ and $\text{top}(v) \in T_w$. But then $\text{top}(v) \in T_u \cap T_w$ implying $uw \in E(G)$, a contradiction. Thus v is indeed a simplicial vertex of G .

Therefore, by Lemma 21, we may assume that $uv \in M$ and $u \in D$ for some vertex $u \in N(v)$. Consider the graph $G' = G - N[u]$, and define $M' = M \setminus \{uv\}$ and $D' = D \setminus \{u\}$. We show that (M', D') is a solution to G' . To see this note first that $N[v] \subseteq N[u]$ and that no neighbour of u or v is an endpoint of an edge in M' , because M is an induced matching and $uv \in M$. This proves that M' is an induced matching of G' . Similarly, note that no neighbour of u belongs to D' , since D is an independent set and $u \in D$. Therefore, D' is an independent set of G' , and clearly, M' contains D' . It remains to show that D' is a maximal independent set of G' . If not, there exists $w \in V(G')$ such that $D' \cup \{w\}$ is an independent set of G' . But then $D' \cup \{u, w\} = D \cup \{w\}$ is an independent set of G , since $(D' \cup \{w\}) \cap N[u] \subseteq V(G') \cap N[u] = \emptyset$. This contradicts the maximality of D , and so we conclude that (M', D') is indeed a solution to G' .

Now, we construct a tree model of G' by taking T as the host tree and the collection $\{T_x\}_{x \in V(G) \setminus N[u]}$ as subtrees of T corresponding to the vertices of G' . By induction, there exists a canonical solution (M'', D'') to G' with respect to this tree model where $|M''| = |M'|$. In particular, every $xy \in M''$ satisfies $x \in D''$ if $\text{top}(x) \succ \text{top}(y)$, and $y \in D''$ if $\text{top}(x) \prec \text{top}(y)$.

Define $M^* = M'' \cup \{uv\}$ and $D^* = D'' \cup \{u\}$. It follows that (M^*, D^*) is a solution to G , since G' contains no neighbour of u and $N[v] \subseteq N[u]$. By Lemma 13 and the minimality of v , we also conclude $\text{top}(v) \preceq \text{top}(u)$. Thus (M^*, D^*) is actually a canonical solution to G , and $|M^*| = |M''| + 1 = |M'| + 1 = |M|$ as required.

That concludes the proof. \square

Proof of Lemma 15. Consider a node $Y \preceq X$ satisfying either the assumption of (i) or the assumption of (ii). Define $M' = M \cap E(G_Y)$ and $D' = D \cap V(G_Y)$. We show that (M', D') is a canonical solution to G_Y which will imply that $Y \in \mathcal{S}$ and the claim will follow.

Since (M, D) is a solution to G_X , it follows that M' is an induced matching of G_Y , and D' is an independent set of G_Y . We show that M' contains D' . Suppose otherwise, namely that there exists $u \in D'$ that is not an endpoint of any edge in M' . Since $u \in D$, there exists v such that $uv \in M$. Thus $v \notin V(G_Y)$ by our assumption about u . Hence, from the definition of G_Y , we obtain $\text{top}(u) \preceq Y$ and $\text{top}(v) \not\preceq Y$. This together with Lemma 13 implies that $\text{top}(v) \succ \text{top}(u)$. But then (M, D) is not a canonical solution to G_X , a contradiction.

It remains to show that D' is a maximal independent set of G_Y . Suppose otherwise, namely that there exists a vertex $v \in V(G_Y)$ non-adjacent to every vertex in D' . Observe that $v \notin D$ by the definition of D' . Recall that (M, D) is a solution to G_X , and hence, D is a maximal independent set of G_X . Thus, there exists $w \in D$ such that $vw \in E(G)$. This implies $w \notin D'$, and so $w \notin V(G_Y)$. Thus, the definition of G_Y yields $\text{top}(v) \preceq Y$ and $\text{top}(w) \not\preceq Y$. Therefore, by Lemma 13, we conclude that $\text{top}(w) \succ \text{top}(v) \in V(T_w)$. This shows that $\text{top}(w) \succeq Y \succeq \text{top}(v)$ which implies $Y \in V(T_w)$, since T_w is a connected subgraph of T and $\text{top}(v) \in V(T_w)$.

We now conclude that Y does not satisfy the assumption of (ii), since $Y \in V(T_w)$ and $w \in D$. So, Y must satisfy the assumption of (i), namely that there exist $u \in D$ with $\text{top}(u) = Y$. We conclude that $u \neq w$, since $\text{top}(w) \succ Y$. Therefore, $uw \in E(G)$, since $\text{top}(u) = Y \in V(T_w)$. But this is impossible, since D is an independent set of G_X and $u, w \in D$, a contradiction.

This proves that $Y \in \mathcal{S}$. Finally, we observe that if Y satisfies the assumption of (i), then the vertex u such that $Y = \text{top}(u)$ also belongs to D' which shows that $Y \in \mathcal{R}$.

That concludes the proof. \square

Proof of Lemma 16. Suppose that there exists a solution (M, D) to G_X . By Lemma 14, we may assume that (M, D) is a canonical solution to G_X . We prove that (M^*, D^*) is also a canonical solution to G_X .

First, observe that for all distinct $Y, Y' \in \mathcal{R}_X^*$, there is no edge in G with one endpoint in G_Y and another endpoint in $G_{Y'}$. Indeed, any such edge uv where $u \in V(G_Y)$ and $v \in V(G_{Y'})$ must satisfy $\text{top}(u) \preceq Y$ and $\text{top}(v) \preceq Y'$. But that contradicts Lemma 13, since Y and Y' are incomparable by \preceq .

This observation shows that M^* is indeed an induced matching, and D^* is an independent set of G_X . Also, if (M^*, D^*) is a solution to G_X , then it is a canonical solution, since (M_Y, D_Y) is a canonical solution to G_Y for all $Y \in \mathcal{R}_X^*$.

Thus, it remains to show that D^* is a maximal independent set of G_X . Suppose otherwise, namely that there exists a vertex $v \in V(G_X)$ non-adjacent to every vertex in D^* .

Consider a node $Y \in \mathcal{R}_X^*$. Observe that $v \notin V(G_Y)$. Indeed, if v is in G_Y , then (M_Y, D_Y) is not a solution to G_Y , since we assume that v is non-adjacent to every vertex of $D^* \supseteq D_Y$. Next, recall that (M_Y, D_Y) is a rooted canonical solution of G_Y . Thus, there exists $w \in D_Y$ with $\text{top}(w) = Y$. Suppose that v is adjacent to some $u \in V(G_Y)$. We conclude $\text{top}(u) \preceq Y$ and $\text{top}(v) \not\preceq Y$ by the definition of G_Y . Hence, by Lemma 13, $\text{top}(v) \succ \text{top}(u) \in V(T_v)$ which implies $\text{top}(v) \succeq \text{top}(w) \succeq \text{top}(u)$. So $\text{top}(w) \in V(T_v)$, since T_v is connected. But now $vw \in E(G)$ and $w \in D^*$ contradicting our assumption that v is not adjacent to any vertex of D^* .

This proves that for every $Y \in \mathcal{R}_X^*$, the vertex v is neither in G_Y nor is adjacent to some vertex in G_Y . Suppose that $v \in D$ and recall that (M, D) is a canonical solution to G_X . This implies by Lemma 15 that $\text{top}(v) \in \mathcal{R}_X$. Thus, since \mathcal{R}_X^* is the set of maximal elements of \mathcal{R}_X with respect to \preceq , there must exist $Y \in \mathcal{R}_X^*$ such that $Y \succeq \text{top}(v)$. Thus $v \in V(G_Y)$ by the definition of G_Y , which is a contradiction to the above. So we conclude that $v \notin D$ and we recall that D is a maximal independent set of G_X . Thus, there exists $u \in D$ such that $uv \in E(G)$.

Similarly as before, we conclude that $\text{top}(u) \in \mathcal{R}_X$ by Lemma 15. Thus, there exists $Y \in \mathcal{R}_X^*$ such that $Y \succeq \text{top}(u)$. Consequently, $u \in V(G_Y)$ by the definition of G_Y . But again this is impossible, since $uv \in E(G)$ and v is not adjacent to any vertex in G_Y as shown above.

This shows that D^* is indeed a maximal independent set of G_X and concludes the proof. \square

Proof of Lemma 17. First, note similarly as in the proof of Lemma 16 that M_{uv}^* is an induced matching and D_{uv}^* is an independent set of G . This follows from the fact that the elements of \mathcal{L}_{uv}^* are pairwise incomparable by \preceq , and hence, the elements in $\bigcup_{X \in \mathcal{L}_{uv}^*} \mathcal{R}_X^*$ are pairwise incomparable by \preceq .

Now, let (M, D) be a canonical solution to G_Z with $uv \in M$ and $u \in D$. For contradiction, suppose that (M_{uv}^*, D_{uv}^*) is not a solution to G_Z . Namely, suppose that there exists a vertex $w \in V(G_Z)$ that is not adjacent to any vertex in D_{uv}^* .

Consider $X \in \mathcal{L}_{uv}^*$, and define $M_X^* = \bigcup_{Y \in \mathcal{R}_X^*} M_Y$ and $D_X^* = \bigcup_{Y \in \mathcal{R}_X^*} D_Y$. Note, by the definition of \mathcal{L}_{uv}^* , that the parent of X belongs to $V(T_u) \cup V(T_v)$. Also, recall that (M, D) is a canonical solution to G_Z with $uv \in M$. Thus X necessarily satisfies one of the assumptions in Lemma 15 which allows us to conclude that $X \in \mathcal{S}$. In other words, there exists a solution to G_X . Thus, by Lemma 16, we conclude that (M_X^*, D_X^*) is also a solution to G_X . In particular, D_X^* is a maximal independent set of G_X . This shows that w is not a vertex of G_X , since $D_{uv}^* \supseteq D_X^*$ and we assume that $D_{uv}^* \cup \{w\}$ is an independent set of G .

We conclude that $\text{top}(w) \not\preceq X$ for all $X \in \mathcal{L}_{uv}^*$. This implies that $\text{top}(w) \notin \mathcal{L}_{uv}$. Recall that $w \in V(G_Z)$ which means $\text{top}(w) \preceq Z$. So $\text{top}(w) \in V(T_u) \cup V(T_v)$ by the definition of \mathcal{L}_{uv} .

Now, recall that (M, D) is a solution to G_Z and so D is a maximal independent set of G_Z . Thus there exist $x \in D$ such that $xw \in E(G)$. Since $u \in D$ but also $u \in D_{uv}^*$, we conclude that $x \neq u$. This implies that $\text{top}(x) \in \mathcal{L}_{uv}$. Hence, there exists $X \in \mathcal{L}_{uv}^*$ such that $\text{top}(x) \preceq X$, and so, by Lemma 15, we conclude that $\text{top}(x) \in \mathcal{R}_X$. Thus there exists $Y \in \mathcal{R}_X^*$ such that $\text{top}(x) \preceq Y$. Recall that (M_Y, D_Y) is a rooted canonical solution to G_Y . Thus, there exists $y \in D_Y$ such that $\text{top}(y) = Y$. Recall that $\text{top}(w) \not\preceq X$ by the above, and so $\text{top}(w) \not\preceq Y$. This implies, by Lemma 13, that $\text{top}(x) \preceq \text{top}(y) \preceq \text{top}(w)$ and $\text{top}(x) \in V(T_w)$. Thus $\text{top}(y) \in V(T_w)$ since T_w is connected, which implies that y is adjacent to w . But then y cannot be in D_Y , since $y \in D_Y \subseteq D_{uv}^*$ and we assume that $D_{uv}^* \cup \{w\}$ is an independent set of G_Z , a contradiction.

Thus no such w exists which proves that D_{uv}^* is indeed a maximal independent set of G_Z . That concludes the proof. \square

Proof of Theorem 3; running time analysis. Let us now analyze the complexity of the algorithm described in the proof of Theorem 3. Let $n = |V(G)|$ and $m = |E(G)|$. We may assume that every connected component of G contains at least one edge, otherwise there is no solution to G . This allows us to assume that $n = O(m)$.

Each time we process a node Z , we test a subset of edges of G_Z . Note that each edge of G is tested this way exactly once during the whole run of the algorithm. To test an edge uv , we construct the sets M_{uv}^* and D_{uv}^* by combining solutions for descendants of Z . To do this efficiently, we augment the algorithm to store a rooted canonical solution to G_Y when it finds one when processing the node Y . Based on this, to construct M_{uv}^* and D_{uv}^* it suffices to search through the descendants of Z in T . This clearly takes at most $O(n)$ time, since both the sets M_{uv}^* and D_{uv}^* have no more than n elements and also T contains at most n nodes. Afterwards, we test if D_{uv}^* is a dominating set of G_Z . This can be done directly in time $O(m)$ by exploring the neighbourhood of every vertex in $V(G_Z) \setminus D_{uv}^*$. The same applies to the construction and testing of M^* and D^* after processing all nodes.

Thus, altogether, the total complexity is $O(m^2)$ and that concludes the proof. \square

Proof of Theorems 7, 8; running time analysis. Let $n = |V(G)|$. To find a solution of size k , we compute k sets $\mathcal{R}^{(1)}, \dots, \mathcal{R}^{(k)}$ each of which has at most n elements.

For $j \in \{1 \dots k\}$, to find out if $Z \in \mathcal{R}^{(j)}$, we apply Lemma 19. This involves choosing integers k_1, \dots, k_t and a set of t nodes, constructing the sets M^* and D^* , and testing if this constitutes a solution to G_Z . There are $\binom{j-t}{t-1} = O(j^t)$ choices for the integers and $O(n^t)$ choices for the nodes. The construction and testing of (M^*, D^*) takes $O(n^2)$ time (assuming we are storing partial solutions just like in the algorithm in Section 5).

Altogether, for all j and all t , the complexity is $\sum_{j=0}^k \sum_{t=0}^{\ell} O(j^t n^{t+2}) = O(k^\ell n^{\ell+2})$ to compute the sets $\mathcal{R}^{(1)}, \dots, \mathcal{R}^{(k)}$. Similar argument yields that the complexity of Lemma 18 applied to the root is $O(k^\ell n^{\ell+2})$ once we have the sets $\mathcal{R}^{(1)}, \dots, \mathcal{R}^{(k)}$.

That completes the proof of Theorem 8. Note that for interval graphs this analysis yields complexity $O(kn^3)$. \square

Proof of Theorem 1. The problem is clearly in NP. To prove that it is also NP-hard we construct a reduction from 3SAT which is NP-hard [9]. Consider an instance \mathcal{I} to 3SAT, namely clauses C_1, \dots, C_m using variables v_1, \dots, v_n . By standard arguments, we may assume that no variable appears twice in the same clause (positively or negatively). We write \bar{v}_i for the negation of v_i .

From the instance, we construct a graph $G_{\mathcal{I}}$ as follows. The vertex set of $G_{\mathcal{I}}$ is $\{c_1, \dots, c_m, v_1, \dots, v_n, \bar{v}_1, \dots, \bar{v}_n\}$. It contains edges $v_1 \bar{v}_1, \dots, v_n \bar{v}_n$ and there is an edge between v_i and c_j if the literal v_i appears in the clause C_j while there is an edge between \bar{v}_i and c_j if \bar{v}_i appears in C_j . Finally, the set $\{c_1, \dots, c_m\}$ forms a clique.

We now prove that \mathcal{I} has a satisfying truth assignment if and only if $G_{\mathcal{I}}$ has an induced matching containing a maximal independent set.

Suppose that \mathcal{I} has a satisfying truth assignment $\varphi : \{v_1, \dots, v_n\} \rightarrow \{\text{true}, \text{false}\}$. Define $M = \{v_1 \bar{v}_1, \dots, v_n \bar{v}_n\}$ and $D = \{v_i \mid \varphi(v_i) = \text{true}\} \cup \{\bar{v}_i \mid \varphi(v_i) = \text{false}\}$. By construction, M is an induced matching containing D , and D is an independent set. It remains to show that D is a maximal independent set of $G_{\mathcal{I}}$. If not, there exist $j \in \{1 \dots m\}$ such that $D \cup \{c_j\}$ is an independent set of $G_{\mathcal{I}}$. This implies that the literals that appear in C_j are not in D , since they are adjacent to c_j . But then they all evaluate to false which means that C_j is not satisfied by the assignment φ , a contradiction.

Conversely, suppose that $G_{\mathcal{I}}$ contains an induced matching M containing an independent set D . Suppose that D contains c_j for some $j \in \{1 \dots m\}$. Suppose that the literal v_i appears in C_j . Recall that we assume that no variable appears both positively and negatively in any clause. Thus c_j is not adjacent to \bar{v}_i . Also, since D is a maximal independent set of $G_{\mathcal{I}}$, either \bar{v}_i or some neighbour of \bar{v}_i belongs to D . Clearly, $v_i \notin D$, since v_i is adjacent to c_j . Also, no neighbour of \bar{v}_i among c_1, \dots, c_m can be in D , since $c_j \in D$ and D is an independent set while $\{c_1, \dots, c_m\}$ is a clique of $G_{\mathcal{I}}$. This yields that $\bar{v}_i \in D$. In particular, M contains an edge whose one endpoint is c_j and another edge whose one endpoint is v_i . But then M cannot be an induced matching, since $N(\bar{v}_i) \subseteq N(c_j)$, a contradiction. We reach a similar contradiction if \bar{v}_i appear in C_j . This proves that $D \cap \{c_1, \dots, c_m\} = \emptyset$ which also shows that for each $i \in \{1 \dots n\}$, either v_i or \bar{v}_i belongs to D but not both. This allows us to define a truth assignment φ as follows. We set $\varphi(v_i)$ to true if $v_i \in D$, and to false otherwise.

We claim that φ is a satisfying truth assignment. Consider a clause C_j . Recall that $D \cap \{c_1, \dots, c_m\} = \emptyset$. So $c_j \notin D$ which means that there is a neighbour of c_j that belongs to D , because D is a maximal independent set. This neighbour can be only one of the literals that appear in C_j , since $D \cap \{c_1, \dots, c_m\} = \emptyset$. Thus φ evaluates this literal to true showing that C_j is satisfied by φ . That completes the proof. \square

Proof of Lemma 20. The proof is by reduction from the exact cardinality dominating set problem in general graphs which is NP-hard, since minimum size dominating set problem is NP-hard[9].

Namely, consider a graph $G = (V, E)$ and an integer k . Construct the following graph G' .

$$\begin{aligned} V(G') &= V \cup \{v_1, v_2, v_3, v_4, v_5, v_6 \mid v \in V\} \\ E(G') &= \{uv_1 \mid uv \in E\} \cup \{u_1v_1 \mid u, v \in V\} \cup \{vv_1, vv_2, vv_4, vv_5, vv_6, v_2v_3 \mid v \in V\} \end{aligned}$$

For simplicity, we define $V_i = \{v_i \mid v \in V\}$ for $i \in \{1, 2, 3, 4, 5, 6\}$.

We start by proving that G' is a chordal graph. We construct an elimination ordering π of G' as follows. We first eliminate the elements of $V_6 \cup V_5 \cup V_4 \cup V_3$ (in any order), then the elements of V_2 , after that the elements of V , and finally the elements of V_1 . We claim that π is a perfect elimination of G' . First, we note that $V_3 \cup V_4 \cup V_5 \cup V_6$ is an independent set, each element of which has exactly one neighbour in G' . Further, V_2 is also an independent set, each element of which has two neighbours, one in V_3 and one in V . Finally, note that V_1 is a clique, and V is an independent set. This proves that π is indeed a perfect elimination ordering of G' .

Now, we prove that G has a dominating set of size k if and only if G' has an independent dominating set of size $4|V| - 2k$.

Suppose that G has a dominating set D of size k . Define $D' = D \cup V_3 \cup \{v_4, v_5, v_6 \mid v \in V \setminus D\}$. We claim that D' is an independent dominating set of G' of size $4|V| - 2k$. First, by the definition, the size of D' is $k + |V| + 3(|V| - k) = 4|V| - 2k$. Further, D' is an independent set. To see this recall that $V \cup V_3$ and $V_3 \cup V_4 \cup V_5 \cup V_6$ are independent set of G' , and note that for each $v \in V$, either $v \in D'$ or $v_4, v_5, v_6 \in D'$ but never both. It remains to prove that D' is a dominating set. First, for each $v \in V$, the vertex v_2 is adjacent to $v_3 \in D'$. Further, if $v \in D$, then the vertices v_4, v_5, v_6 are adjacent to $v \in D'$. If $v \notin D$, then v is adjacent to $v_4, v_5, v_6 \in D'$. Finally, for each $u \in V$, there is $v \in D$ with $uv \in E$, since D is a dominating set of G . Thus, u_1 is adjacent to v in G' . This exhausts all vertices of G' thus proving that D' is an independent dominating set of G' .

Conversely, suppose that G' contains an independent dominating set D' of size $4|V| - 2k$. Let $D = D' \cap V$. We prove that D is a dominating set of G of size k . First, observe that if $v \in D$, then V' necessarily contains the vertex v_3 and none of v_2, v_4, v_5, v_6 , because D' is a maximal independent set. Similarly, if $v \notin D$, then D' must contain the vertices v_4, v_5, v_6 and exactly one of v_2, v_3 . This allows us to conclude that $D' \setminus V_1$ contains exactly $4|V| - 2|D|$ vertices.

In fact, this also proves that $D' \cap V_1 = \emptyset$. Indeed, if for some $u \in V$, the vertex u_1 belongs to D' , then no other vertex of V_1 belongs to D' as V_1 is a clique. This implies that the size of D' is $4|V| - 2|D| + 1$. But this is an odd number, while by our assumption, D' has $4|V| - 2k$ vertices, for some k , which is an even number, a contradiction. Thus $D' \cap V_1 = \emptyset$ as claimed.

So, we have that D' is of size exactly $4|V| - 2|D|$ which is $4|V| - 2k$ by our assumption. Thus we conclude that D is exactly of size k . It remains to prove that D is a dominating set of G . Let $u \in V$. If $u \notin D$, then there exists a vertex in D' adjacent to u_1 , since D' is a dominating set in G' . As the vertices in V_1 are only adjacent to the vertices in $V \cup V_1$ and $D' \cap V_1 = \emptyset$, we conclude that some $v \in V$ is adjacent to u_1 in G' . So, by the construction of G' , we have that $uv \in E$. Thus D is indeed a dominating set of G .

That concludes the proof. □

Proof of Theorem 4. We prove Theorem 4 by reduction from the exact cardinality dominating set problem in chordal graphs (Lemma 20). Consider a chordal graph $G = (V, E)$ and an integer k . We construct a graph G' from G by substituting a clique of size two for every vertex of G . Namely, G' is defined as follows.

$$V(G') = V \cup \{v_1 \mid v \in V\}$$

$$E(G') = \{vv_1 \mid v \in V\} \cup \{uv, uv_1, u_1v, u_1v_1 \mid uv \in E\}$$

G' is a chordal graph, since chordal graphs are closed under clique substitution.

Next we verify that G has an independent dominating set of size k if and only if G' has an induced matching of size k containing a maximal independent set.

Suppose that G has an independent dominating set D . Define $M = \{vv_1 \mid v \in D\}$. Clearly, M is an induced matching of G' , since D is an independent set of G . Also, D is contained in M . Further, D is a maximal independent set of G' , because it is an independent dominating set of G . Thus G' contains an induced matching M of size k containing a maximal independent set.

Conversely, suppose that G' contains an induced matching M of size k containing a maximal independent set D . Note that up to renaming the vertices (u_1 to u and u to u_1 for $u \in V$) if necessary, we may assume that $D \subseteq V$. We show that D is an independent dominating set of G of size k . Clearly, D is an independent set, by definition, and $|D| = k$, since $|M| = k$. If D is not a dominating set of G , then there is $v \in V$ such that $D \cup \{v\}$ is an independent set of G . But then $D \cup \{v\}$ is also an independent set of G' contradicting the maximality of D in G' .

That ends the proof. □