Algorithmica (2024) 86:1428–1474
https://doi.org/10.1007/s00453-023-01196-y

Domination and Cut Problems on Chordal Graphs with Bounded Leafage

Esther Galby[1] · Dániel Marx[2] · Philipp Schepper[2] · Roohani Sharma[3] · Prafullkumar Tale[4]

Received: 5 January 2023 / Accepted: 24 November 2023 / Published online: 29 December 2023

Abstract
The leafage of a chordal graph G is the minimum integer ℓ such that G can be realized as an intersection graph of subtrees of a tree with ℓ leaves. We consider structural parameterization by the leafage of classical domination and cut problems on chordal graphs. Fomin, Golovach, and Raymond [ESA 2018, Algorithmica 2020] proved, among other things, that Dominating Set on chordal graphs admits an algorithm running in time $2^{\mathcal{O}(\ell^2)} \cdot n^{\mathcal{O}(1)}$. We present a conceptually much simpler algorithm that runs in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$. We extend our approach to obtain similar results for Connected Dominating Set and Steiner Tree. We then consider the two classical cut problems MultiCut with Undeletable Terminals and Multiway Cut with Undeletable Terminals. We prove that the former is W[1]-hard when parameterized by the leafage and complement this result by presenting a simple $n^{\mathcal{O}(\ell)}$-time algorithm. To our surprise, we find that Multiway Cut with Undeletable Terminals on chordal graphs can be solved, in contrast, in $n^{\mathcal{O}(1)}$-time.

Keywords Chordal graphs · Leafage · FPT algorithms · Dominating set · MultiCut with undeletable terminals · Multiway cut with undeletable terminals

1 Introduction

The intersection graph of a family $\mathcal{F}$ of nonempty sets is the graph whose vertices are the elements of $\mathcal{F}$ with two vertices being adjacent if and only if their corresponding

✉ Esther Galby
esther.galby@tuhh.de

1 Hamburg University of Technology, Hamburg, Germany

2 CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

3 Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

4 Indian Institute of Science Education and Research Pune, Pune, India

sets intersect. The most natural and famous example of such intersection graphs are *interval graphs* where $\mathcal{F}$ is a collection of subpaths of a path. Due to their applicability in scheduling, interval graphs have received a considerable attention in the realm of algorithmic graph theory. One useful characterization of an interval graph is that its maximal cliques can be linearly ordered such that for every vertex, the maximal cliques containing that vertex occur consecutively [26]. This property proves very useful for the design of polynomial-time dynamic programming based or greedy algorithms on interval graphs.

Consider the generalization where $\mathcal{F}$ is a collection of subtrees of a tree instead of subpaths of a path. In this case, the corresponding class of intersection graphs is exactly that of *chordal graphs* [11, 25, 48]. Recall that a graph is chordal if every cycle of length at least 4 has a chord. Often, the algorithms of the types mentioned in the previous paragraph fail to generalize to this superclass as witnessed by the following problems that admit polynomial-time algorithms on interval graphs but are NP-complete on chordal graphs: Dominating Set [8, 13], Connected Dominating Set [3, 49], Steiner Tree [3, 49], Multicut with Undeletable Terminals [29, 45], Subset Feedback Vertex Set (Subset FVS) [22, 46], Longest Cycle [28, 35],[1] Longest Path [33], Component Order Connectivity [20], s-Club Contraction [27], Independent Set Reconfiguration [5], Bandwidth [37], Cluster Vertex Deletion [36]. Also, Graph Isomorphism on chordal graphs is polynomial-time equivalent to the problem on general graphs whereas it admits a linear-time algorithm on interval graphs [41].

The problems above remain hard even on *split graphs*, another well-studied subclass of chordal graphs. A graph is a split graph if its vertex set can be partitioned into a clique and an independent set. The collection of split graphs is a (proper) subset of the class of intersection graphs where $\mathcal{F}$ is a collection of substars of a star. As interval graphs are intersection graphs of subpaths of a path (a tree with two leaves) and split graphs are intersection graphs of substars of a star (a tree with arbitrary number of leaves), a natural question to consider is what happens to these problems on subclasses of chordal graphs that are intersection graphs of subtrees of a tree with a bounded number of leaves. Motivated by such questions, we consider the notion of *leafage* introduced by Lin et al. [40]: the leafage of a chordal graph G is the minimum integer ℓ such that G can be realized as an intersection graph of a collection $\mathcal{F}$ of subtrees of a tree that has ℓ leaves. Note that the leafage of interval graphs is at most 2 while split graphs have unbounded leafage. Thus the leafage measures, in some sense, how close a chordal graph is to an interval graph. Alternately, an FPT or XP algorithm parameterized by the leafage can be seen as a generalization of the algorithm on interval graphs.

Related Work Habib and Stacho [30] showed that we can compute the leafage of a connected chordal graph in polynomial time. Their algorithm also constructs a corresponding *representation tree*[2] T with the minimum number of leaves. In recent years, researchers have studied the structural parameterization of various graph problems on chordal graphs parameterized by the leafage. Fomin et al. [21] and Arvind et al.

[1] See Exercise 2 in Chapter 6 in [28].

[2] We present formal definitions of the terms used in this section in Sect. 2.

[2] proved, respectively, that the Dominating Set and Graph Isomorphism problems on chordal graphs are FPT parameterized by the leafage. Barnetson et al. [4] and Papadopoulos and Tzimas [47] presented XP-algorithms running in time $n^{\mathcal{O}(\ell)}$ for Fire Break and Subset FVS on chordal graphs, respectively. Papadopoulos and Tzimas [47] also proved that Subset FVS is W[1]-hard when parameterized by the leafage. Hochstättler et al [32] showed that we can compute the neighborhood polynomial of a chordal graph in $n^{\mathcal{O}(\ell)}$-time.

It is known that the size of an *asteroidal set* in a chordal graph is upper bounded by its leafage [40]. See [1, 31] for the relationship between leafage and other structural properties of chordal graphs. Kratsch and Stewart [38] proved that we can effectively 2ℓ-approximate bandwidth of chordal graphs of leafage ℓ. Chaplick and Stacho [14] generalized the notion of leafage to *vertex leafage* and proved that, unlike leafage, it is hard to determine the optimal vertex leafage of a given chordal graph. Figueiredo et al. [18] proved that Dominating Set, Connected Dominating Set and Steiner Tree are FPT on chordal graphs when parameterized by the size of the solution plus the vertex leafage, provided that a tree representation with optimal vertex leafage is given as part of the input.

Our Results We consider well-studied domination and cut problems on chordal graphs. As our first result, we prove that Dominating Set on chordal graphs of leafage at most ℓ admits an algorithm running in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$. This improves upon the existing algorithm by Fomin et al. [21, Theorem 9] which runs in time $2^{\mathcal{O}(\ell^2)} \cdot n^{\mathcal{O}(1)}$. Despite being significantly simpler than the algorithm in [21], our algorithm in fact solves the Red- Blue Dominating Set problem, a well-known generalization of Dominating Set. In this generalized version, an input is a graph G with a partition (R, B) of its vertex set and an integer k, and the objective is to find a subset D of R that dominates every vertex in B, i.e., $B \subseteq N(D)$. We further use this algorithm to solve other related domination problems.

Theorem 1.1 Dominating Set, Connected Dominating Set, *and* Steiner Tree *can be solved in* $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$ *on chordal graphs of leafage at most* ℓ.

The reductions in [8] and [49] used to prove that these problems are NP-complete on chordal graphs imply that these problems do not admit $2^{o(n)}$, and hence $2^{o(\ell)} \cdot n^{\mathcal{O}(1)}$, algorithms unless the ETH fails.

Arguably, the two most studied cut problems are MultiCut and Multiway Cut. In the MultiCut problem, an input is a graph G, a set of terminal pairs $P \subseteq V(G) \times V(G)$ and an integer k, and the objective is to find a subset $S \subseteq V(G)$ of size at most k such that no pair of vertices in P is connected in $G - S$. In the Multiway Cut problem, instead of terminal pairs, we are given a terminal set P and the objective is to find a subset $S \subseteq V(G)$ of size at most k such that no two vertices in P are connected in $G - S$. These problems and variations of them have received a considerable attention which lead to the development of new techniques [9, 15, 16, 42, 43]. Misra et al. [44] studied the parameterized complexity of these problems on chordal graphs. Guo et al. [29] proved that MultiCut with Deletable Terminals is NP-complete on interval graphs, thereby implying that this problem is paraNP-hard when parameterized by the leafage. We consider the MultiCut with Undeletable Terminals problem and prove the following result.

Theorem 1.2 Multicut with Undeletable Terminals *on chordal graphs is* W[1]-hard *when parameterized by the leafage ℓ and assuming the* ETH*, does not admit an algorithm running in time $f(\ell) \cdot n^{o(\ell)}$ for any computable function f. However, it admits an* XP*-algorithm running in time $n^{\mathcal{O}(\ell)}$.*

Next, we focus on the Multiway Cut with Undeletable Terminals problem. We find it somewhat surprising that the classical complexity of this problem on chordal graphs was not known. Bergougnoux et al. [7], using the result in [21], proved that the problem admits an XP-algorithm when parameterized by the leafage.[3] Our next result significantly improves upon this and [44, Theorem 2] which states that the problem admits a polynomial kernel when parameterized by the solution size.

Theorem 1.3 Multiway Cut with Undeletable Terminals *can be solved in $n^{\mathcal{O}(1)}$-time on chordal graphs.*

A well-known trick to convert an instance of Multiway Cut with Deletable Terminals into an instance of Multiway Cut with Undeletable Terminals is to add a pendant vertex to each terminal, remove that vertex from the set of terminals, and make the newly added vertex a terminal. As this reduction converts a chordal graph into another chordal graph, Theorem 1.3 implies that Multiway Cut with Deletable Terminals is also polynomial-time solvable on chordal graphs. Another closely related problem is Subset FVS which is NP-complete on split graphs [46]. To the best of our knowledge, this is the first graph class on which the classical complexity of these two problems differ.

Next, we revisit the problems on chordal graphs with bounded leafage and examine how far we can generalize this class. An *asteroidal triple* of a graph G is a set of three vertices such that each pair is connected by some path that avoids the closed neighborhood of the third vertex. Lekkerkerker and Boland [39] showed that a graph is an interval graph if and only if it is chordal and does not contain an asteroidal triple. They also listed all minimal chordal graphs that contain an asteroidal triple (see, for instance, [12, Figure 1]). Among this list, we found the *net graph* to be the most natural to generalize. For a positive integer $\ell \geq 3$, we define H_ℓ as a split graph on 2ℓ vertices with split partition (C, I) such that the only edges across C, I are a perfect matching. Note that H_3 is the net graph. As interval graphs are a proper subset of the collection of chordal graphs that do not contain a net graph as an induced subgraph, the collection of the chordal graph of leafage ℓ is a proper subset of the collection of chordal graphs that do not contain $H_{\ell+1}$ as an induced subgraph (see Observation 6.1). We show that, although the considered domination problems are polynomial-time solvable for constant ℓ, the fixed-parameter tractability results are unlikely to extend to this larger class. Let us mention that the core reason these problems admit XP-algorithms parameterized by ℓ lies in the fact that H_ℓ-induced-subgraph-free chordal graphs have mim-width at most $\ell - 1$ [34] (all three problems are indeed known to be solvable in $n^{\mathcal{O}(m)}$ on graphs of mim-width at most m [6, 10]). Nonetheless, we present alternative algorithms which we believe to be simpler and more insightful. In fact, we give a $n^{\mathcal{O}(\ell)}$ algorithm for the more general Red- Blue Dominating Set problem and obtain the other results by simple reductions.

[3] See the discussion after Corollary 2 on page 1388 in [7].

Theorem 1.4 Dominating Set, Connected Dominating Set *and* Steiner Tree *on H_ℓ-induced-subgraph-free chordal graphs are* W[1]-hard *when parameterized by ℓ and assuming the* ETH, *do not admit an algorithm running in time $f(\ell) \cdot n^{o(\ell)}$ for any computable function f. However, they all admit* XP*-algorithms running in time $n^{\mathcal{O}(\ell)}$.*

We observe a similar trend with respect to MultiCut with Undeletable Terminals as its parameterized complexity jumps from W[1]-hard on chordal graph of leafage ℓ to paraNP-hard on H_ℓ-induced-subgraph-free chordal graphs when parameterized by ℓ.

Theorem 1.5 MultiCut with Undeletable Terminals *is* NP-hard *even when restricted to H_3-induced-subgraph-free chordal graphs.*

Table 1 summarizes our results.
Our Methods We briefly discuss the methods used in our two main algorithms, namely the algorithm for Dominating Set and the one for Multiway Cut.

Red-Blue Dominating Set in Chordal Graphs As mentioned earlier, the linear ordering of cliques in interval graphs is particularly useful for the design of polynomial-time algorithms. Such an ordering is not possible even if G is a chordal graph whose representation tree T is a star. Consider the case where the model of every red vertex in G includes the center of the star T (and possibly some leaves) and the model of every blue vertex is (only) a leaf. We can solve this instance by converting it to an instance of Set Cover and solving it using the FPT algorithm parameterized by the size of the universe. In this case, the size of the universe is at most the number of leaves which is upper bounded by the leafage. In the other case where the properties of red vertices and blue vertices are reversed, we obtain a similar result by creating an equivalent instance of Hitting Set.

These ideas can be used in a more general setting as long as the following two properties are satisfied: (1) the model of each vertex is *local*, that is, it contains at most one branching node, and (2) each branching node is contained only in models of either red vertices or blue vertices. Based on this observation, we introduce a restricted version of the problem in which the input graph is required to satisfy these two conditions. We then show that the general case reduces to this restricted version: indeed, we prove that there is a branching algorithm that constructs $2^{\mathcal{O}(\ell)}$ many instances (where ℓ is the leafage of the input graph) of the restricted version of the problem such that the input instance is a Yes-instance if and only if one of these newly created instances is a Yes-instance. These two properties ensure that the graph induced by the red and blue vertices whose model intersect the subtree rooted at a farthest branching node (from some fixed root) satisfies the premise of at least one of the cases mentioned in the previous paragraph. We then present a greedy procedure, based on solving the Set Cover and Hitting Set problems, that identifies some part of an optimum solution. Apart from this greedy selection procedure, all other steps of the algorithm run in polynomial time.

Multiway Cut in Chordal Graphs We give a polynomial-time algorithm for Multiway Cut on chordal graphs by solving several instances of the (s, t)- Cut problem

Table 1 Overview of the known results and our contributions

Input graph	Dom Set, ConnDom Set, Steiner Tree	MultiCut with UnDel Term	MultiwayCut
Interval Graphs	Poly-time [3, 13]	Poly-time [29]	Poly-time [7]
Chordal graphs of leafage ℓ	$2^{\mathcal{O}(\ell^2)} \cdot n^{\mathcal{O}(1)}$ algo [21] $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$ algo (Thm 1.1)	W[1]-hard $n^{\mathcal{O}(\ell)}$ algo (Thm 1.2)	$n^{\mathcal{O}(\ell)}$ algo [7] Poly-time (Thm 1.3)
H_ℓ-induced subgraph-free chordal	W[1]-hard (Thm 1.4); $n^{\mathcal{O}(\ell)}$ algo (Thm 1.4, [34] + [6, 10])	NP-hard for $\ell \geq 3$ (Thm 1.5)	Poly-time (Thm 1.3)
Chordal graphs	NP-complete [8]	NP-complete [49]	Poly-time (Thm 1.3)

Every graph class mentioned in the first column is a proper subset of the graph class mentioned below (not necessarily with unit capacities). Our strategy is based on a bottom-up dynamic programming (DP) on a tree representation of a chordal graph. An interesting aspect of our DP is that we need to look-up *all* DP table values that are already computed to compute a new entry. This is in contrast to typical DP-based algorithms that do computations only based on *local* entries.

We remark that we do not expect to design an algorithm for Multiway Cut on chordal graphs using much simpler arguments (like a simple dynamic programming procedure etc.) as the problem generalizes some well-studied cut-flow based problems. As an example, recall the Vertex Cover problem on bipartite graphs where given a bipartite graph G with bipartition (A, B), the goal is to find $A' \subseteq A$ and $B' \subseteq B$ such that $|A' \cup B'|$ is minimum and $N(A \setminus A') \subseteq B'$. The set $A' \cup B'$ is called a vertex cover of G. The Vertex Cover problem on bipartite graphs reduces to the Multiway Cut problem on chordal graphs: indeed, let G' be the graph obtained from G by making B a clique, adding new pendant vertex t_a to each vertex $a \in A$, and further adding another new vertex t that is adjacent to all vertices of B. Then G' is a chordal graph and letting $T = t \cup \{t_a \mid a \in A\}$, it is easy to see that $S \subseteq V(G)$ is a vertex cover of G if and only if S is a T-multiway-cut in G'. As mentioned earlier, our algorithm solves several instances of the (s, t)- Cut problem, which also sits at the heart of some algorithms for Vertex Cover on bipartite graphs. The above reduction suggests that an algorithm for Multiway Cut on chordal graphs using much simpler techniques, would imply an algorithm for Vertex Cover on bipartite graphs that uses much simpler techniques as well.

Note that a similar reduction would work from the weighted variant of the Vertex Cover problem on bipartite graphs. This can be achieved by further replacing each vertex of the graph G by a clique of size proportional to the weight of this vertex and making each vertex of the clique adjacent to all the neighbors of this vertex. This reduction still preserves the chordality of the resulting graph.

Organization of the Paper In Sect. 2, we define the notations and terminology used throughout the paper. In Sect. 3, we present the FPT algorithm for the generalized Red- Blue Dominating Set problem parameterized by the leafage. In Sect. 4, we consider the Multicut problem and provide the proof of Theorem 1.2. We present the polynomial-time algorithm for Multiway Cut on chordal graphs in Sect. 5. In Sect. 6, we revisit the aforementioned problems by restricting the input to H_ℓ-induced-subgraph-free chordal graphs and prove Theorem 1.4 for Dominating Set as well as Theorem 1.5. Finally in Sect. 7, we consider the Connected Dominating Set and the Steiner Tree problems and complete the proofs of Theorem 1.1 and Theorem 1.4.

2 Preliminaries

For a positive integer q, we denote the set $\{1, 2, \dots, q\}$ by $[q]$ and for any $0 \leq p \leq q$, we denote the set $\{p, \dots, q\}$ by $[p, q]$. We use $\mathbb{N}$ to denote the set of all non-negative integers. Given a function $f : X \to Z$ and $Y \subseteq X$, $f|_Y$ denotes the function f restricted to Y.

Graph Theory For a graph G, we denote by $V(G)$ and $E(G)$ the set of vertices and edges of G, respectively. Unless specified otherwise, we use n to denote the number of vertices in G. We denote the edge with endpoints u, v by uv. For any $v \in V(G)$, $N_G(v) = \{u \mid uv \in E(G)\}$ denotes the (open) neighbourhood of v, and $N_G[v] = N_G(v) \cup \{v\}$ denotes the closed neighbourhood of v. When the graph G is clear from the context, we omit the subscript G. For any $S \subseteq V(G)$, $G - S$ denotes the graph obtained from G by deleting vertices in S. We denote the subgraph of G induced by S, i.e., the graph $G - (V(G) \setminus S)$, by $G[S]$. We say graph G contains graph H as in *induced subgraph* if H can be obtained from G by series of vertex-deletions. Recall that for a directed graph H, we denote by $N_H^+(v)$ the out-neighbors of $v \in V(H)$ and by $N_H^-(v)$ the in-neighbors of $v \in V(H)$. If H is clear from the context, we omit the subscript H. Given a (directed) path P in a graph G and two vertices $u, v \in V(P)$, we denote by $P[u, v]$ the subpath of P from u to v. For any further notation from basic graph theory, we refer the reader to [19].

Trees A *tree* T is a connected acyclic graph. Consider a tree T rooted at r. We define function $\texttt{parent}(t, T) : V(T) \setminus \{\mathsf{r}\} \mapsto V(T)$ to specify unique parent of the nodes in rooted tree T. For any node $t \in T$, we denote by T_t the subtree rooted at t. A *subdivided star* is a tree with at most one vertex of degree at least 3 (in other words, it is a tree obtained by repeatedly subdividing the edges of a star graph). The sets $V_{\geq 3}(T)$ and $V_{=1}(T)$ denote the set of vertices of degree at least 3, and of degree equal to 1, respectively. The set $V_{\geq 3}(T)$ is also called the set of *branching vertices* of T and the set $V_{=1}(T)$ is called the set of *leaves* of T. Note that $|V_{\geq 3}(T)| \leq |V_{=1}(T)| - 1$. Any node of T which is not a leaf is called *internal*.

Chordal Graphs and Tree Representations A graph is called a chordal graph if it contains no induced cycle of length at least four. It is well-known that chordal graphs can be represented as intersection graphs of subtrees in a tree, that is, for every chordal graph G, there exists a tree T and a collection $\mathcal{M}$ of subtrees of T in one-to-one correspondence with $V(G)$ such that two vertices in G are adjacent if and only if their corresponding subtrees intersect. The pair $(T, \mathcal{M})$ is called a *tree representation* of G. For every $v \in V(G)$, we denote by $\mathcal{M}(v)$ the subtree corresponding to v and refer to $\mathcal{M}(v)$ as the *model* of v in T. Throughout this article, we use *nodes* to refer to the vertices of the tree T to avoid confusion with the vertices of the graph G. Furthermore, we use the greek alphabet to denote nodes of T and the latin alphabet to denote vertices of G. For notational convenience, for any node $\alpha \in V(T)$ and edge $e \in E(T)$, we may abuse notation and write $\alpha \in \mathcal{M}(v)$ in place of $\alpha \in V(\mathcal{M}(v))$ as well as $e \in \mathcal{M}(v)$ in place of $e \in E(\mathcal{M}(v))$.

For every node $\alpha \in V(T)$, we let $\texttt{ver}(\alpha) = \{v \in V(G) \mid \alpha \in \mathcal{M}(v)\}$, that is, $\texttt{ver}(\alpha)$ is the set of vertices in G that contain the node α is their model. A vertex $v \in V(G)$ whose model contains α may also be referred to as an *α-vertex*. Similarly, for every edge $e \in E(T)$, we let $\texttt{ver}(e) = \{v \in V(G) \mid e \subseteq \mathcal{M}(v)\}$, that is, $\texttt{ver}(e)$ is the set of vertices of G that contain the edge e in their model. Given a subtree T' of T, we denote by $G_{|T'}$ the subgraph of G induced by those vertices $x \in V(G)$ such that $V(\mathcal{M}(x)) \subseteq V(T')$. If T is rooted, then for each vertex $v \in V(G)$, we call the node in $\mathcal{M}(v)$ that is closest to the root of T, the *topmost* node of $\mathcal{M}(v)$ and denoted it by $\texttt{top}_{\mathcal{M}}(v)$.

The *leafage* of chordal graph G, denoted by $\mathtt{lf}(G)$, is defined as the minimum number of leaves in the tree of a tree representation of G. A tree representation $(T, \mathcal{M})$ for G such that the number of leaves in T is $\mathtt{lf}(G)$, can be computed in time $O(|V(G)|^3)$ [30]. Furthermore, the number of nodes in T is at most $O(|V(G)|)$.

Parameterized Complexity The input of a parameterized problem comprises an instance I, which is an input of the classical instance of the problem, and an integer k, which is called the parameter. A parameterized problem Π is said to be *fixed-parameter tractable* (FPT for short) if given an instance (I, k) of Π, we can decide whether (I, k) is a YES-instance of Π in time $f(k) \cdot |I|^{\mathcal{O}(1)}$ for some computable function f depending only on k. We say that an instance (I, k) of a parameterized problem Π and an instance (I', k') of a parameterized problem Π' (possibly $\Pi = \Pi'$) are *equivalent* if $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi'$. A *reduction rule*, for a parameterized problem Π, is a polynomial-time algorithm that takes as input an instance (I, k) of Π and outputs an instance (I', k') of Π. If (I, k) and (I', k') are equivalent then we say that the reduction rule is *safe*. For more details on parameterized algorithms, and in particular parameterized branching algorithms, we refer the reader to the book by Cygan et al. [17].

3 Dominating Set

For a graph G, a set $X \subseteq V(G)$ is a *dominating set* if every vertex in $V(G) \setminus X$ has at least one neighbor in X, that is, $V(G) = N[X]$. In the DOMINATING SET problem (DOMSET for short), the input is a graph G and an integer k, and the objective is to decide whether G has a dominating set of size at most k. We assume that the leafage of the input graph is given as part of the input. If not, recall that it can be computed in polynomial time [30]. We consider a generalized version of this problem as defined below.

RED- BLUE DOMINATING SET (RED- BLUE- DOMSET)
Input: A graph G, a partition (R, B) of $V(G)$, and an integer k.
Question: Does there exist a set $X \subseteq R$ of size at most k such that $B \subseteq N(X)$?

We first prove that to solve DOMSET, it is sufficient to solve RED- BLUE- DOMSET even when the input is restricted to chordal graphs of leafage ℓ. There is indeed a simple reduction from the former problem to the latter that preserves the properties in which we are interested.

Lemma 3.1 *There is a polynomial-time algorithm that given an instance (G, k) of* DOMSET *constructs an equivalent instance $(G', (R', B'), k)$ of* RED- BLUE- DOMSET *such that if G has leafage at most ℓ, then so does G'.*

Proof We construct G' from G as follows. For every vertex $v \in V(G)$, add two copies v_R and v_B to $V(G')$ and add an edge $v_R v_B$ to $E(G')$. For every edge $uv \in E(G)$, add edges $v_R u_R$, $v_R u_B$, $v_B u_R$, and $v_B u_B$ to $E(G')$. This completes the construction of G'. Let $R' = \{v_R \mid v \in V(G)\}$ and $B' = \{v_B \mid v \in V(G)\}$.

Suppose that the DOMSET instance has a solution $S \subseteq V(G)$. Then the set $S_R = \{v_R \mid v \in S\}$, i.e., S_R contains the red version of each vertex in S, is a solution for the RED- BLUE- DOMSET instance: indeed, the blue vertices v_B such that $v \notin S$ are dominated since S is a solution, and if $v \in S$, then v_B is dominated because of the newly added edges. Conversely, if $S_R \subseteq R'$ is a solution for the RED- BLUE- DOMSET instance, then it is easy to see that $S = \{v \mid v_R \in S_R\}$ is a solution for the DOMSET instance.

Finally note that a tree representation for G' can be obtained from a tree representation for G by duplicating the model of each vertex, and making the original model a model for the blue version of the vertex, and the copy a model for its red version. In particular, the leafage of G' is at most that of G. □

In the remainder of this section, we present an FPT algorithm for RED- BLUE- DOMSET when parameterized by the leafage ℓ of the input graph. The algorithm consists of two parts. In the first part, the algorithm constructs $2^{\mathcal{O}(\ell)}$ many instances of a "restricted version" of the problem such that the input instance is a YES-instance if and only if one of these newly created instances is a YES-instance. Moreover, the graphs in the newly created instances satisfy certain properties that allow us to design a fast algorithm. See Lemma 3.2 for the formal statement. In the second part (cf. Lemma 3.3), the algorithm solves the restricted version of RED- BLUE- DOMSET which is defined as follows.

RESTRICTED- RED- BLUE DOMINATING SET (REST- RED- BLUE- DOMSET)
Input: A chordal graph G, a partition (R, B) of $V(G)$, an integer k and tree representation $(T, \mathcal{M})$ of G such that

- for every vertex in G, its model contains at most one branching node of T, and
- for all branching nodes $\gamma \in V(T)$, there are either only red γ-vertices or only blue γ-vertices.

Question: Does there exist a set $D \subseteq R$ of size at most k such that $B \subseteq N(D)$?

3.1 Constructing REST-RED-BLUE-DOMSET Instances

In this section, we prove the following result.

Lemma 3.2 *Let $\mathcal{I} = (G, (R, B), k)$ be an instance of* RED- BLUE- DOMSET *where G is a chordal graph of leafage at most ℓ. We can construct, in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$, a collection $\{\mathcal{I}_i = (G_i, (R_i, B_i), k) \mid i \in [2^{\mathcal{O}(\ell)}]\}$ of* REST- RED- BLUE- DOMSET *instances such that*

- *For every $i \in [2^{\mathcal{O}(\ell)}]$, G_i is a chordal graph of leafage at most 2ℓ, and*
- *$\mathcal{I}$ is a* YES*-instance of* RED- BLUE- DOMSET *if and only if at least one of the instances in the collection is a* YES*-instance of* REST- RED- BLUE- DOMSET.

Proof Let G be a chordal graph and let $(T, \mathcal{M})$ be a tree representation of G. We define the following functions.

- Let $f_T(G)$ denote the number of branching nodes $\gamma \in V(T)$ such that there exist both a red vertex and a blue vertex whose models contain γ.

- Let $f_r(G)$ denote the number of pairs of consecutive branching nodes α, β in T (that is, no node on the unique path in T from α to β is a branching node) such that there is red vertex whose model contains both α and β.
- Similarly, let $f_b(G)$ denote the number of pairs of consecutive branching nodes α, β in T such that there is blue vertex whose model contains both α and β.

We further define $\mu(G) := \mathtt{lf}(G) + 2 \cdot (f_T(G) + f_r(G) + f_b(G))$. Note that, by definition, $\mu(G) \geq \mathtt{lf}(G)$. We design a polynomial-time branching algorithm whose measure μ decreases in each branch. We first show that if $\mu(G) = \mathtt{lf}(G)$ then $(G, (R, B), k)$ is in fact an instance of Rest- Red- Blue- DomSet and then show how the branching algorithm proceeds.

Assume therefore that $\mu(G) = \mathtt{lf}(G)$. Then $f_T(G) = f_r(G) = f_b(G) = 0$ by definition. However, when $f_T(G) = 0$, then, by definition, for every branching node $\gamma \in V(T)$, all the vertices containing γ in their model are either red or blue; and when $f_r(G) = f_b(G) = 0$ then, considering the fact that every model is a subtree in T, for every vertex in G, its model contains at most one branching node in T. Therefore if $\mu(G) = \mathtt{lf}(G)$, then $(G, (R, B), k)$ is also an instance of Rest- Red- Blue- DomSet.

Now assume that $\mu(G) > \mathtt{lf}(G)$. Then $f_T(G) + f_r(G) + f_b(G) > 0$. We consider the following three exhaustive cases.

Case I $f_T(G) > 0$. Let γ be a branching node in T such that there is both a red-vertex and a blue-vertex whose models contain γ. Suppose that $\mathcal{I}$ is a Yes-instance of Red- Blue- DomSet and let D be a solution. Consider first the case where D includes a red vertex whose model contains γ. In this case, we return the instance $\mathcal{I}_1 = (G_1, (R_1, B_1), k)$ which is obtained as follows.

- Initialize $V(G_1) = V(G)$, $R_1 = R$, $B_1 = B$.
- Let T_1 be the tree obtained from T by adding a node δ and making it adjacent to γ only. Note that $V(T_1) \setminus \{\delta\} \subseteq V(T)$.
- For every red vertex $v \in V(G_1)$ such that $\gamma \in \mathcal{M}(v)$, add δ to its model, i.e., $\mathcal{M}_1(v) = \mathcal{M}(v) \cup \{\delta\}$.
- For every blue vertex $v \in V(G_1)$ such that $\gamma \in \mathcal{M}(v)$, delete v from $V(G_1)$.
- Add a new blue vertex x to $V(G_1)$ and to B_1 with $\mathcal{M}_1(x) = \{\delta\}$.
- For every (red or blue) vertex $v \in V(G)$ such that $\gamma \notin \mathcal{M}(v)$, define $\mathcal{M}_1(v_1) = \mathcal{M}(v)$.

It is easy to verify that $(T_1, \mathcal{M}_1)$ is a tree representation of G_1 and that T_1 has exactly one more leaf than T, i.e., $\mathtt{lf}(G_1) \leq \mathtt{lf}(G) + 1$. However, since we have deleted all the blue vertices whose models contained γ, $f_T(G_1) = f_T(G) - 1$. As the other parts of the measure do not change, $\mu(G_1) < \mu(G)$.

In the second case where no vertex in D contains γ in its model, we return the instance $\mathcal{I}_2 = (G_2, (R_2, B), k)$ where G_2, R_2 are obtained from G, R, respectively, by deleting red vertices whose model contains γ. It is easy to verify that $\mu(G_2) < \mu(G)$.

If $\mathcal{I}$ is a Yes-instance, then at least one of $\mathcal{I}_1$ or $\mathcal{I}_2$ is a Yes-instance as these two branches are exhaustive. If $\mathcal{I}_1$ is a Yes-instance, then any optimum solution must include a red γ-vertex because of the newly added vertex x. As $R_2 \subseteq R$, if $\mathcal{I}_2$ is a Yes-instance, then $\mathcal{I}$ is a Yes-instance. Hence, this branching step is correct.

Case II $f_T(G) = 0$ and $f_r(G) > 0$. Let α, β be two consecutive branching nodes in T such that there is a red vertex whose model contains both α and β. Suppose that $\mathcal{I}$ is a Yes-instance of Red- Blue- DomSet and let D be a solution. Consider the case where D includes a red vertex whose model contains both α and β. In this case, we return the instance $\mathcal{I}_1 = (G_1, (R_1, B_1), k)$ which is obtained as follows.

- Initialize $V(G_1) = R_1 = B_1 = \emptyset$.
- Let T_1 be the tree obtained from T by contracting the unique path $P_{\alpha\beta}$ from α to β in T and let $\gamma_{\alpha\beta}$ be the node resulting from this contraction. Add a node δ to T_1 and make it adjacent to $\gamma_{\alpha\beta}$ only. Note that $V(T_1) \setminus \{\gamma_{\alpha\beta}, \delta\} \subseteq V(T)$.
- For every red vertex $v \in V(G)$ such that $\mathcal{M}(v) \cap V(P_{\alpha\beta}) \neq \emptyset$, add a red vertex v_1 to $V(G_1)$ (and to R_1) with $\mathcal{M}_1(v_1) = (\mathcal{M}(v) \setminus V(P_{\alpha\beta})) \cup \{\gamma_{\alpha\beta}, \delta\}$.
- Add a new blue vertex x to $V(G_1)$ with $\mathcal{M}_1(x) = \{\delta\}$.
- For every (red or blue) vertex $v \in V(G)$ such that $\mathcal{M}(v) \cap V(P_{\alpha\beta}) = \emptyset$, add v_1 to G_1 (and to, respectively, either R_1 or B_1) with $\mathcal{M}_1(v_1) = \mathcal{M}(v)$.

Note that for every blue vertex $v \in V(G)$ such that $\mathcal{M}(G) \cap V(P_{\alpha\beta}) \neq \emptyset$, there is no corresponding blue vertex in G_1. It is easy to verify that $(T_1, \mathcal{M}_1)$ is a tree representation of G_1 and that T_1 has one more leaf than T which implies $\texttt{lf}(G_1) \leq \texttt{lf}(G) + 1$. Since we have contracted the path $P_{\alpha\beta}$ to obtain the node $\gamma_{\alpha\beta}$, $f_r(G_1) < f_r(G)$. As the other parts of the measure do not change, $\mu(G_1) < \mu(G)$.

In the second case where no vertex in D contains both α and β in its model, we return an instance $\mathcal{I}_2 = (G_2, (R_2, B), k)$ where G_2, R_2 are obtained from G, R, respectively, by deleting red vertices whose model contains both α and β. It is easy to verify that $\mu(G_2) < \mu(G)$. We argue as in the previous case for the correctness of this branching steps.

Case III $f_T(G) = 0$ and $f_b(G) > 0$. Let α, β be two consecutive branching nodes in T such that there is a blue vertex whose model contains both α and β. Note that since $f_T(G) = 0$, for every red vertex $v \in V(G)$ such that $\mathcal{M}(v) \cap V(P_{\alpha\beta}) \neq \emptyset$, in fact $\mathcal{M}(v) \subseteq V(P_{\alpha\beta}) \setminus \{\alpha, \beta\}$. Suppose that $\mathcal{I}$ is a Yes-instance of Red- Blue- DomSet and let D be a solution. Consider first the case where D includes a red vertex whose model is in $V(P_{\alpha\beta}) \setminus \{\alpha, \beta\}$. In this case, we return the instance $\mathcal{I}_1 = (G_1, (R, B_1), k)$ where G_1, B_1 are obtained from G and B as follows.

- Delete all the blue vertices whose model contains both α and β.
- Add a blue vertex x to $V(G_1)$ (and to B_1) with $\mathcal{M}(x) = V(P_{\alpha\beta}) \setminus \{\alpha, \beta\}$.

It is easy to verify that $(T, \mathcal{M})$ is a tree representation of G_1 and $f_b(G_1) < f_b(G)$. As the other parts of the measure do not change, $\mu(G_1) < \mu(G)$.

In the second case where there is no vertex in D whose model is in $V(P_{\alpha\beta}) \setminus \{\alpha, \beta\}$, we consider the following two subcases. If there is a blue vertex v such that $\mathcal{M}(v) \subseteq V(P_{\alpha\beta})$, then we return a trivial No-instance. Otherwise, we return the instance $\mathcal{I}_2 = (G_2, (R_2, B_2), k)$ which is constructed as follows.

- Initialize $V(G_2) = R_2 = B_2 = \emptyset$.
- Let T_2 be the tree obtained from T by contracting the path $P_{\alpha\beta}$ from α to β in T and let $\gamma_{\alpha\beta}$ be the node resulting from this contraction. Note that $V(T_2) \setminus \{\gamma_{\alpha\beta}\} \subseteq V(T)$.
- For every (red or blue) vertex $v \in G$ such that $\mathcal{M}(v) \cap V(P_{\alpha\beta}) = \emptyset$, add a vertex v_2 to G_2 (and to, respectively, either R_2 or B_2) with $\mathcal{M}_2(v_2) = \mathcal{M}(v)$.

- For every blue vertex $v \in V(G)$ such that $\mathcal{M}(v) \cap V(P_{\alpha\beta}) \neq \emptyset$, add a blue vertex v_2 to $V(G_2)$ (and to B_2) with $\mathcal{M}_2(v_2) = (\mathcal{M}(v_2) \setminus V(P_{\alpha\beta})) \cup \{\gamma_{\alpha\beta}\}$.

Note that for any red vertex $v \in V(G)$ such that $\mathcal{M}(v) \subseteq V(P_{\alpha\beta})\setminus\{\alpha, \beta\}$, there is no corresponding red vertex in G_2. It is easy to verify that $(T_2, \mathcal{M}_2)$ is a tree representation of G_2. Furthermore, the number of leaves of T_2 is the same as T and $f_b(G_2) < f_b(G)$. As the other parts in the measure do not change, $\mu(G_2) < \mu(G)$.

The correctness of this branching step follows from the same arguments as in the previous cases and the fact that in the second case, since there is no red vertex whose model intersects $V(P_{\alpha\beta})$, it is safe to contract that path.

Finishing the Proof The correctness of the overall algorithm follows from the correctness of branching steps in the above three cases. To bound its running time and the number of instances it outputs, note that $f_T(G) + f_r(G) + f_b(G) \leq 3 \cdot \texttt{lf}(G)$ as these functions either count the number of branching nodes or the unique paths containing exactly two (consecutive) branching nodes. ☐

3.2 Solving an Instance of Rest-Red-Blue-DomSet

In this section, we present an algorithm to solve Rest- Red- Blue- DomSet. Formally, we prove the following lemma.

Lemma 3.3 Rest- Red- Blue- DomSet *admits an algorithm running in time* $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$.

We first state some easy reduction rules before we handle two cases based on whether the farthest branching node[4] is contained only in the models of red vertices or blue vertices. We present Greedy Select 3.9 and Greedy Select 3.12 to handle these cases. The proof of the lemma follows from Lemma 3.10, Lemma 3.13 and the fact that each application of the greedy selection procedure deletes some vertices in the graph.

We first introduce some notations. Recall that an instance of Rest- Red- Blue- DomSet contains a chordal graph G, a partition (R, B) of $V(G)$, an integer k and tree representation $(T, \mathcal{M})$ of G such that for every vertex in G, its model contains at most one branching node of T, and for all branching nodes $\gamma \in V(T)$, there are either only red γ-vertices or only blue γ-vertices. We assume, without loss of generality, that the tree T is rooted at node r. Unless mentioned otherwise, α denotes the farthest branching node in T from the root, that is, each proper subtree of T_α is a path. If there are more than one branching node that satisfy the property, we arbitrarily select one of them. Let β be the closest branching ancestor of α, that is, no internal node in the unique path from α to β is a branching node in T.[5] Recall that for a vertex $v \in V(G)$, we define $\texttt{top}_{\mathcal{M}}(v)$ as the node $\eta \in \mathcal{M}(v)$ that is closest to the root. Likewise if a leaf λ is fixed, we define $\texttt{bot}^{\lambda}_{\mathcal{M}}(v)$ as the node $\eta \in \mathcal{M}(v)$ that is closest to λ. For ease of notation, we omit λ as it is always clear from the context.

[4] We assume that the tree in the tree representation is rooted and thus, by farthest branching node, we mean farthest from the root.

[5] If α is the root of the tree, then we can add an artificial new root β which is not contained in the model of any vertex.

Definition 3.4 Let γ be a node of the tree T. We define the following sets of vertices in G.

- $B_\gamma^\cap$, $R_\gamma^\cap$, $V_\gamma^\cap$ are the sets of, respectively, blue, red, all vertices $v \in V(G)$ whose models intersect the tree rooted at γ, i.e., $\mathcal{M}(v) \cap V(T_\gamma) \neq \emptyset$.
- $B_\gamma^\subseteq$, $R_\gamma^\subseteq$, $V_\gamma^\subseteq$ are the sets of, respectively, blue, red, all vertices $v \in V(G)$ whose models are completely contained inside the tree rooted at γ, i.e., $\mathcal{M}(v) \subseteq V(T_\gamma)$.
- $B_\gamma^{\subseteq\dagger}$, $R_\gamma^{\subseteq\dagger}$, $V_\gamma^{\subseteq\dagger}$ are the sets of blue, red, all vertices $v \in V(G)$ where the model is completely contained inside the tree rooted at γ but does not contain γ, respectively, i.e. $\mathcal{M}(v) \subseteq V(T_\gamma^\dagger) = V(T_\gamma) \setminus \{\gamma\}$.
- $B_\gamma^\in$, $R_\gamma^\in$, $V_\gamma^\in$ are the sets of, respectively, blue, red, all vertices $v \in V(G)$ whose models contains γ, i.e., $\gamma \in \mathcal{M}(v)$.

Simplifications We first apply the following easy reduction rules whose correctness readily follows from the definition of the problem. It is also easy to see that the reduction rules can be applied in polynomial time and the reduced instance is also a valid instance of Rest- Red- Blue- DomSet.

Reduction Rule 3.5 *If there is a blue vertex, which is not adjacent to a red vertex, or if $k < 0$, then return a trivial* No-*instance.*

Reduction Rule 3.6

- *If there are two blue vertices u, v such that $\mathcal{M}(u) \subseteq \mathcal{M}(v)$, then delete v.*
- *If there are two red vertices u, v such that $\mathcal{M}(u) \subseteq \mathcal{M}(v)$, then delete u.*

Consider a blue vertex v in G whose model is contained in the subtree rooted at α. Moreover, let v be such a vertex for which $\texttt{top}_{\mathcal{M}}(v)$ is farthest from the root and v is not adjacent to a red vertex whose model contains α. Hence, there is a natural ordering amongst the red neighbors of v. Note that such an ordering is not possible if some of its neighbors contain α in their models. As any solution contains a red neighbor of v, it is safe to include its neighbor v_r for which $\texttt{top}_{\mathcal{M}}(v_r)$ is closest to α.

Reduction Rule 3.7 *Suppose that there is a blue vertex $v \in B_\alpha^{\subseteq\dagger}$ such that $\texttt{top}_{\mathcal{M}}(v)$ is farthest from the root and v is not adjacent to any red α-vertices. Moreover, amongst all the red neighbors of v, let v_r be the node such that $\texttt{top}_{\mathcal{M}}(v_r)$ is closest to α. Then, remove v_r and all of its blue neighbors and decrease k by* 1.

We remark that the above reduction rule is applicable irrespective of the fact whether either all α-vertices are red or all α-vertices are blue.

Case 1: All the vertices that contain α in their models are red vertices Let β be the closest branching ancestor of α. Consider the blue vertices whose model intersect the path from α to β. Note that there may not be any such blue vertex; however, we find it convenient to present an uniform argument. With a slight abuse of notation, let $b_1, \dots, b_d$ be these blue vertices ordered according to their endpoint in the direction of α, that is, for $i < j$ we have either $\texttt{bot}_{\mathcal{M}}(b_i) = \texttt{bot}_{\mathcal{M}}(b_j)$ or $\texttt{bot}_{\mathcal{M}}(b_i)$ is closer to α than $\texttt{bot}_{\mathcal{M}}(b_j)$. For each $i \in [d]$, we compute an optimal solution for dominating the vertices whose model is in the tree rooted at α (i.e., the vertices of

$B_\alpha^{\subseteq\dagger}$) and the vertex b_i while only using red α-vertices. Formally, we want to compute an optimal solution for the following instance: $\mathcal{I}_i := G[R_\alpha^{\cap} \cup B_\alpha^{\subsetneq} \cup \{b_i\}]$. We also define instance $\mathcal{I}_0 := G[R_\alpha^{\cap} \cup B_\alpha^{\subsetneq}]$ to handle the cases when there are no blue vertices whose model intersects the path from α to β or when b_1 (and hence, the other blue vertices mentioned above) are not dominated by red α-vertices in an optimum solution. To simplify notation we set $\text{OPT}_i := \text{OPT}(\mathcal{I}_i)$ in the following. If $\mathcal{I}_i$ is not defined, then we set $\text{OPT}_i = \infty$. Note that the solution OPT_i also dominates the blue vertices $b_1, \ldots, b_{i-1}$ due to the ordering of the b_is. Hence, for any $i, j \in [0, d]$ such that $i < j$, we have $|\text{OPT}_i| \leq |\text{OPT}_j|$. We use this monotonicity to prove the following structural lemma.

Lemma 3.8 *Let $q \in [0, d]$ be the largest value such that $|\text{OPT}_q| = |\text{OPT}_0|$. If there is a solution, then there is an optimum solution containing OPT_q.*

Proof Let OPT be an optimum solution of $(G, (R, B), k)$. Let S denote the collection of vertices in OPT whose model contains nodes in the subtree rooted at α, i.e., $S := \text{OPT} \cap R_\alpha^{\cap}$. We claim that we can replace S by a super-set S' of OPT_q of equal size to obtain another solution.

Let $j \in [0, d]$ be the largest integer such that b_j is dominated by some vertex in S. If $j \leq q$, then by our choice of q, $|S| = |\text{OPT}_q|$. By the definition of the $\mathcal{I}_i$s, we get that OPT_q is also a solution for $\mathcal{I}_i$. Hence, we can replace S by OPT_q to get another optimal solution. Suppose therefore that $j > q$. By our choice of q, we have $|S| > |\text{OPT}_q|$. Let r_j be the red α-vertex with $\mathtt{top}_{\mathcal{M}}(r_j)$ closest to β such that b_j is a neighbor of r_j. Such a vertex exists, as by assumption, S contains one of these vertices which dominates b_j. Then we replace S by $S' = \text{OPT}_q \cup \{r_j\}$. As $|S| > |\text{OPT}_q|$, we have $|S'| \leq |S|$. Moreover, observe that $S' \cup \text{OPT} \setminus S$ is still a solution as all vertices in $B_\alpha^{\subseteq\dagger}$ and the vertices $b_1, \ldots, b_q$ are dominated by some vertex in OPT_q, vertex r_j dominates the vertices $b_{q+1}, \ldots, b_j$ and, by the choice of j, the vertices $b_{j+1}, \ldots, b_d$ are dominated by some vertex not contained in S. □

We devise a greedy selection step based on the above lemma.

Greedy Select 3.9 *Let $q \in [0, d]$ be the largest value such that $|\text{OPT}_q| = |\text{OPT}_0|$. Include the vertices of OPT_q in the solution, i.e., delete the red vertices in OPT_q, the blue vertices that are adjacent to vertices in OPT_q, and decrease k by $|\text{OPT}_q|$.*

Lemma 3.10 *Greedy Select3.9 step is correct and can be completed it time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$.*

Proof The correctness of the step follows directly from Lemma 3.8. In the remaining proof, we show how to compute, for every $i \in [0, d]$, OPT_i in time $\mathcal{O}(\ell \cdot |R| \cdot 2^\ell \cdot n)$ by constructing an instance of Set Cover. Before constructing such an instance, we justify that only one blue vertex (which is farthest from α) is critical while constructing this Set Cover instance.

Let α' be a child of α. As α is a farthest branching node of T from the root, the tree rooted at α' is a path. Let λ be the another endpoint of this path. Consider a blue vertex $v_{\alpha'}$ whose model is contained in $T_{\alpha'}$, i.e., $v_{\alpha'} \in B_{\alpha'}^{\subseteq}$. Moreover, suppose that $v_{\alpha'}$ is the vertex for which $\mathtt{top}_{\mathcal{M}}(v_r)$ is farthest from α'. As Reduction Rule 3.7 is not applicable, there exists at least one red neighbor of $v_{\alpha'}$ which is an α-vertex. Hence, an optimum solution can always include a red neighbor of $v_{\alpha'}$ which is also an α-vertex. This red α-vertex also dominates all the blue vertices in $B^{\subsetneq}_{\alpha'}$.

We now explain how to construct an instance $(U, \mathcal{F})$ of Set Cover. For every child α' of α, if the vertex $v_{\alpha'}$ mentioned in the previous paragraph exists, then add an element $u_{\alpha'}$ corresponding to it to U. When $i \neq 0$, add another element u_i corresponding to b_i to U. For every red α-vertex v, we define set $S_v \subseteq U$ as the collection of elements corresponding to the blue vertices in $\mathcal{I}_i$ that are adjacent to v. This completes the construction of the instance.

It is easy to see the one-to-one correspondence between the optimum solutions of these two instances. The running time of the algorithm follows from the known algorithms for Set Cover (see, for instance, [23]) and the fact that α has at most ℓ many children. □

Case 2: All the vertices that contain α in their models are blue vertices Let β be the closest branching ancestor of α. We consider two cases depending on whether there is a red vertex whose model intersects the path from α to β. If there is no such red vertex, then we consider the graph induced by all the red vertices whose model is (properly) contained in the subtree rooted at α and the blue vertices whose model intersects the subtree rooted at α. Formally, we define $\mathcal{I}_0 = G[R^{\subsetneq}_{\alpha} \cup B^{\cap}_{\alpha}]$.

Consider the other case and suppose that there are $d \geq 1$ many red vertices whose model intersects the path from α to β. Let $r_1, \ldots, r_d$ be these vertices ordered according to their endpoints in the direction of α, that is, for $i < j$, we have either $\texttt{bot}_{\mathcal{M}}(r_i) = \texttt{bot}_{\mathcal{M}}(r_j)$ or $\texttt{bot}_{\mathcal{M}}(r_i)$ is closer to α than $\texttt{bot}_{\mathcal{M}}(r_j)$. For each such red vertex v_i, we compute the optimal solution to dominate the vertices in $B^{\cap}_{\alpha}$ by vertices in $R^{\subsetneq}_{\alpha}$ assuming that v_i is already selected. Note that we only have to focus on the blue vertices in $B^{\cap}_{\alpha}$ which are not adjacent to v_i. Formally we define $\mathcal{I}_i = G[R^{\subsetneq}_{\alpha} \cup (B^{\cap}_{\alpha} \setminus N[v_i])]$. It is possible that the optimum solution does not include any of the vertices in $\{r_1, r_2, \ldots, r_d\}$. To handle this case, we define $\mathcal{I}_{d+1} = G[R^{\subsetneq}_{\alpha} \cup B^{\cap}_{\alpha}]$. To simplify notation, we set $\text{OPT}_i := \text{OPT}(\mathcal{I}_i)$ in the following. Note that for the instance defined above, R_i is same for every instance whereas $B_i \subseteq B_{i+1}$ because of the ordering. Hence, for any $i, j \in [d+1]$ such that $i < j$, we have $|\text{OPT}_i| \leq |\text{OPT}_j|$. We use this monotonicity to prove the following structural lemma.

Lemma 3.11 *If there is a red vertex whose model intersects the path from* α *to* β, *let* $q \in [d+1]$ *be the largest value such that* $|\text{OPT}_q| = |\text{OPT}_1|$. *Otherwise, define* $\text{OPT}_q = \text{OPT}_0$. *If there is a solution for the instance, then there is an optimum solution* OPT *such that* $\text{OPT} \cap R^{\subsetneq}_{\alpha} = \text{OPT}_q$.

Proof If there is no red vertices whose model intersects the path from α to β, then all the red vertices in G that are adjacent to blue vertices in $\mathcal{I}_0$ are the red vertices in $\mathcal{I}_0$. Hence, the statement of the lemma follows.

We now consider the case where there are red vertices whose model intersects the path from α to β. Let OPT be an optimum solution of $(G, (R, B), k)$. Let S denote the collection of vertices in OPT whose model is (properly) contained in the subtree rooted at α, i.e., $S := \text{OPT} \cap R^{\subsetneq\dagger}_{\alpha}$. We claim that we can replace S by a super-set S' of OPT_q of equal size to obtain another optimum solution.

Let $j \in [d]$ be the smallest index such that v_j is contained in OPT. Note that, by definition, $j \neq d+1$ as there are only d red vertices with the said property. If $j \leq q$, then by our choice of q, $|S| \geq |\mathrm{OPT}_j|$. By the definition of $\mathcal{I}_j$ and the fact blue vertices in $\mathcal{I}_j$ are subset of blue vertices in $\mathcal{I}_q$, OPT_q is also a solution for $\mathcal{I}_j$. Hence, we can replace S by OPT_q to get another optimal solution. Suppose therefore that $j > q$. By our choice of q, we have $|\mathrm{OPT}_j| > |\mathrm{OPT}_q|$. As OPT is a solution, all vertices in $B_\alpha^{\cap}$ must be covered by OPT. Hence, we can replace S by $S' = \mathrm{OPT}_q \cup \{r_q\}$ and get a solution of not larger size which still dominates all vertices in $B_\alpha^{\cap}$. Indeed, the vertices which are not dominated by OPT_q are dominated by r_q. □

We devise a greedy selection step based on the above lemma.

Greedy Select 3.12 *If there is a red vertex whose model intersects the path from α to β, let $q \in [d+1]$ be the largest value such that $|\mathrm{OPT}_q| = |\mathrm{OPT}_1|$. Otherwise, define $\mathrm{OPT}_q = \mathrm{OPT}_0$. Include OPT_q in the solution, i.e., delete the red vertices in OPT_q, the blue vertices that are adjacent to vertices in OPT_q, and decrease k by $|\mathrm{OPT}_\ell|$.*

Lemma 3.13 *Greedy Select 3.12 step is correct and can be completed in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$.*

Proof The correctness of the step follows directly from Lemma 3.11. In the remaining proof, we show how to compute OPT_i for every $i \in [0, d+1]$, by constructing an instance of Hitting Set. As in Lemma 3.10, we first argue that only one red vertex (which is closest to α) is critical while constructing a Hitting Set instance.

Recall that, by assumption, none of the previous reduction rules are applicable. As in the previous case, let α' be a child of α. We first argue that there are no blue vertices whose path is completely contained in the path rooted at α'. Assume, for the sake of contradiction, that there exists such a blue vertex v. As Reduction Rule 3.5 is not applicable, v is adjacent to at least one red vertices. However, since all α-vertices are blue, by the property of the instance, there are no red α-vertices. In particular, v is not adjacent to any red α-vertex. This contradicts the fact that there exists such a blue 3.7 is not applicable. Hence, there is no blue vertex whose model is contained in the path rooted at α'. Since this is true for any child of α, there are no blue vertices in $B_\alpha^{\subseteq\dagger}$, i.e., $B_\alpha^{\subseteq\dagger} = \emptyset$ and $B_\alpha^{\cap} = B_\alpha^{\in}$. Now, for a child α' of α, let $v_{\alpha'} \in R_{\alpha'}^{\subseteq}$ be a red vertex such that $\mathtt{top}_{\mathcal{M}}(v_r)$ is closest to α. Since all blue vertices contain α in their model, the only critical red vertex in this leg is $v_{\alpha'}$.

We now explain how to construct an instance $(U, \mathcal{F})$ of Hitting Set. For every child α' of α, let $v_{\alpha'}$ be the vertex as mentioned above. Add an element $u_{\alpha'}$ corresponding to $v_{\alpha'}$ in U. For every blue α-vertex v, we define $S_v \subseteq U$ as the collection of elements corresponding to the red vertices in $\mathcal{I}_i$ that are adjacent to v. This completes the construction of the instance.

It is easy to see the one-to-one correspondence between the optimum solutions of these two instances. The running time of the algorithm follows from the simple brute-force algorithm for Hitting Set parameterized by the size of the universe and the fact that α has at most ℓ many children. □

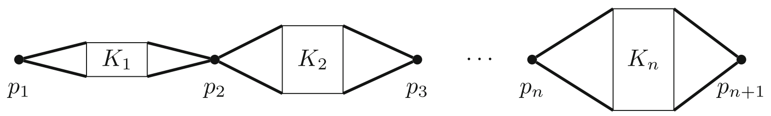


Fig. 1 The auxiliary graph B. Rectangles represent cliques and thick edges indicate that the corresponding vertex is complete to the corresponding cliques

4 Multicut with Undeletable Terminals

This section considers the MULTICUT WITH UNDELETABLE TERMINALS problem formally defined as follows.

MULTICUT WITH UNDELETABLE TERMINALS (MULTICUT WITH UNDEL TERM)
Input: An undirected graph G, a set $P \subseteq V(G) \times V(G)$, and an integer k.
Question: Is there a set $S \subseteq V(G) \setminus V(P)$ such that $|S| \leq k$ and for all $(p, p') \in P$, there is no path between p and p' in $G - S$?

In the following, a set $S \subseteq V(G) \setminus V(P)$ such that for all $(p, p') \in P$, there is no path between p and p' in $G - S$ is called a *P-multicut* in G. We first prove that when the input is restricted to chordal graphs, the problem is unlikely to admit an FPT algorithm when parameterized by the leafage. We then complement this result with an XP-algorithm parameterized by the leafage. We restate the theorem with the precise statement for the reader's convenience.

Theorem 1.2 MULTICUT WITH UNDELETABLE TERMINALS on chordal graphs is *W[1]-hard* when parameterized by the leafage ℓ and assuming the *ETH*, does not admit an algorithm running in time $f(\ell) \cdot n^{o(\ell)}$ for any computable function f. However, it admits an XP-algorithm running in time $n^{\mathcal{O}(\ell)}$.

To prove that the problem is W[1]-hard, we present a parameter preserving reduction from MULTICOLORED CLIQUE. An instance of this problem consists of a simple graph G, an integer q, and a partition $(V_1, V_2, \ldots, V_q)$ of $V(G)$. The objective is to determine whether there is a clique in G that contains exactly one vertex from each part V_i. Such a clique is called a *multicolored clique*. We assume, without loss of generality, that each V_i is an independent set and that $|V_1| = \ldots = |V_q| = n$. [6] This implies, in particular, that $|E(G)| < n^2 \cdot q^2$. For every $i \in [q]$, we denote by $v^i_1, \ldots, v^i_n$ the vertex set of V_i and for every $i \neq j \in [q]$, we denote by $E_{i,j} \subseteq E(G)$ the set of edges between V_i and V_j. We define $M := (n + 1)^2 \cdot q^2$.

Reduction The reduction takes as input an instance $(G, q, (V_1, \ldots, V_q))$ of MULTICOLORED CLIQUE and outputs an instance (H, P, k) of MULTICUT WITH UNDEL TERM which is constructed as follows.

- The reduction starts by constructing an auxiliary graph B. The vertex set of B consists of $n + 1$ vertices $p_1, \ldots, p_{n+1}$ and n vertex-disjoint cliques $K_1, \ldots, K_n$ such that $|K_a| = a \cdot M$ for every $a \in [n]$. Then, it adds edges so that p_1 is complete

[6] Unlike in the rest of the article, we *do not* use n to denote the total number of vertices in G to keep notation simple while presenting the reduction.

to K_1, p_{n+1} complete to K_n, and p_a complete to $K_{a-1} \cup K_a$ for every $a \in [n] \setminus \{1\}$. This completes the construction of B (see Fig. 1).

- For each $i \in [q]$, the reduction introduces two vertex-disjoint copies $B^{i,\alpha}$ and $B^{i,\beta}$ of B. For every $i \in [q]$, let $p_1^{i,\alpha}, \ldots, p_{n+1}^{i,\alpha}$ denote the copies of $p_1, \ldots, p_{n+1}$ in $B^{i,\alpha}$ and $K_1^{i,\alpha}, \ldots, K_n^{i,\alpha}$ denote the copies of $K_1, \ldots, K_n$ in $B^{i,\alpha}$. Moreover, for every $1 \leq a_1 \leq a_2 \leq n+1$, we define, for notational convenience,

$$p^{i,\alpha}[a_1, a_2] := \{p_a^{i,\alpha} \mid a_1 \leq a \leq a_2\}$$

and

$$K^{i,\alpha}[a_1, a_2] := \bigcup_{a_1 \leq a \leq a_2} K_a^{i,\alpha}.$$

We define $p_a^{i,\beta}$, $K_a^{i,\beta}$, $p^{i,\beta}[a_1, a_2]$, and $K^{i,\beta}[a_1, a_2]$ in a similar way.

- For $i \in [q]$ and $a \in [n]$, the reduction uses $p_a^{i,\alpha}$, $p_{n+1-a}^{i,\beta}$, $K_a^{i,\alpha}$, and $K_{n+1-a}^{i,\beta}$ to encode vertex v_a^i.
- For every edge $e = v_{a_i}^i v_{a_j}^j \in E(G)$, the reduction introduces an *edge-vertex* v_e and adds edges so that v_e is complete to the following sets.
  - $p^{i,\alpha}[a_i + 1, n+1]$ and $K^{i,\alpha}[a_i, n+1]$ in $V(B^{i,\alpha})$.
  - $p^{j,\alpha}[a_j + 1, n+1]$ and $K^{i,\alpha}[a_j, n+1]$ in $V(B^{j,\alpha})$.
  - $p^{i,\beta}[n+1-a_i+1, n+1]$ and $K^{i,\beta}[n+1-a_i+1, n+1]$ in $V(B^{i,\beta})$.
  - $p^{j,\beta}[n+1-a_j+1, n+1]$ and $K^{i,\beta}[n+1-a_j+1, n+1]$ in $V(B^{j,\beta})$.

  Note that v_e is adjacent to vertices in $K^{i,\alpha}[a_i] \cup K^{j,\alpha}[a_j]$ but not to any vertex in $K^{i,\beta}[n+1-a_i] \cup K^{j,\beta}[n+1-a_j]$.
- The reduction introduces a central clique K of size $2M^2$ and makes it complete to $\{p_{n+1}^{i,\alpha}, p_{n+1}^{i,\beta} \mid i \in [q]\}$ and V_E where $V_E = \{v_e \mid e \in E(G)\}$ is the set of edge-vertices. This completes the construction of H.
- The reduction further defines

$$P := \{(p_a^{i,\alpha}, p_{n+2-a}^{i,\beta}) \mid a \in [n] \text{ and } i \in [q]\}, \text{ and}$$
$$k := q(n+1)M + |E(G)| - q(q-1)/2.$$

The reduction returns (H, P, k) as the instance of MultiCut with UnDel Term. This completes the reduction. It is easy to see that H is chordal and has leafage at most $2q$. See Fig. 2 for a tree representation of H.

Intuition We first provide the intuition behind the reduction. Recall that the reduction uses $p_a^{i,\alpha}$, $p_{n+1-a}^{i,\beta}$, $K_a^{i,\alpha}$, and $K_{n+1-a}^{i,\beta}$ to encode vertex v_a^i where $i \in [q]$ and $a \in [n]$. Hence, for $a, b \in [n]$, if $a + b = n + 1$, then $p_a^{i,\alpha}$ and $p_b^{i,\beta}$ correspond to the same vertex. Note that the pairs in P *do not* correspond to the vertices associated with v_a^i. Rather, $p_{a+1}^{i,\alpha}$ is paired with $p_{n+1-a}^{i,\beta}$. Conversely, for $a, b \in [n]$, if $a + b = n + 2$, then

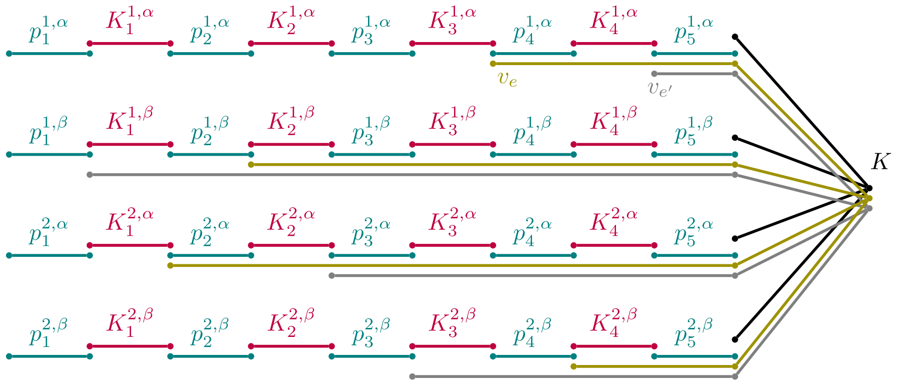


Fig. 2 A tree representation of the graph H restricted to the gadgets representing V_1, V_2 and $E_{1,2}$ where $n = 4$ and $E_{1,2} = \{e = v_3^1 v_1^2, e' = v_4^1 v_2^2\}$

$(p_a^{i,\alpha}, p_b^{i,\beta}) \in P$. By the construction of H and P, for a P-multicut S of H, if there is a path from $p_a^{i,\alpha}$ to $p_b^{i,\beta}$ in $H - S$, then $a + b \geq n + 3$.

Now, consider the terminal pairs $(p_1^{i,\alpha}, p_{n+1}^{i,\beta})$ in P for some $i \in [q]$. Because of the size constraints, S cannot contain all the vertices of the central clique K. Since S cannot contain a terminal, it needs to include one clique from $B^{i,\alpha}$. Let $a_i \in [n]$ be the largest index such that $K_{a_i}^{i,\alpha} \subseteq S$. Using similar arguments, there must also exist $b_i \in [n]$ such that $K_{b_i}^{i,\beta} \subseteq S$ and b_i is largest such index. By definition of a_i, b_i and construction of H, there is a path from $p_{a_i+1}^{i,\alpha}$ to $p_{b_i+1}^{i,\beta}$ in $H - S$. The discussion in the previous paragraph implies that $a_i + 1 + b_i + 1 \geq n + 3$, i.e., $a_i + b_i \geq n + 1$. However, by definition of the solution size k and the size of the cliques, we have $a_i + b_i \leq n + 1$. Hence, the structure of the auxiliary graphs and the terminal pairs ensure that the selected cliques in $S \cap V(B^{i,\alpha})$ and $S \cap V(B^{i,\beta})$ encode selecting a vertex in V_i in G.

Suppose that $\{v_{a_1}^1, v_{a_2}^2, \ldots, v_{a_q}^q\}$ are the vertices in G that are selected by S. Recall that V_E is the collection of edge-vertices in H. Considering the remaining budget, a solution S can include at most $|E(G)| - q(q-1)/2$ many vertices in V_E. We argue that $q(q-1)/2$ edges in G corresponding to vertices in $V_E \setminus S$ should have their endpoints in $\{v_{a_1}^1, v_{a_2}^2, \ldots, v_{a_q}^q\}$ as otherwise some terminal pair is connected in $H - S$. Hence, a P-multicut S of H corresponds to a multicolored clique in G. We formalize this intuition in the following two lemmas.

Lemma 4.1 *If $(G, q, (V_1, V_2, \ldots, V_q))$ is a* Yes*-instance of* Multicolored Clique*, then (H, P, k) is a* Yes*-instance of* MultiCut with UnDel Term.

Proof Assume that $(G, q, (V_1, V_2, \ldots, V_q))$ is a Yes-instance of Multicolored Clique and let $\{v_{a_1}^1, v_{a_2}^2, \ldots, v_{a_q}^q\}$ be a clique in G such that $v_{a_i}^i \in V_i$ for every $i \in [q]$. We construct a P-multicut S of H as follows. First, we add $V_E \setminus \{v_e \mid e \in \{v_{a_i}^i v_{a_j}^j \mid i, j \in [q]\}\}$ to S. For every $i \in [q]$, we further add $K_{a_i}^{i,\alpha}$ and $K_{n+1-a_i}^{i,\beta}$ to S. It is easy to verify that $|S| = q(n+1)M + |E(G)| - q(q-1)/2 = k$.

Let us show that S is indeed a P-multicut. Fix indices $i \in [q]$ and $a \in [n]$, and consider the terminal pair $(p_a^{i,\alpha}, p_{n+2-a}^{i,\beta})$ in P. Suppose first that $a \leq a_i$. By construction of H, any path from $p_a^{i,\alpha}$ to $p_{n+2-a}^{i,\beta}$ in H contains a vertex of $K_{a_i}^{i,\alpha}$ or of $N(p_{a_i}^{i,\alpha}) \cap V_E$. Recall that if edge $e \in E(G)$ is incident on $v_{a_i}^i$, then the edge-vertex v_e in H is adjacent only to vertices in $p^{i,\alpha}[a_i+1, \cdots, n+1]$ and $K^{i,\alpha}[a_i, \cdots, n+1]$ in $V(B^{i,\alpha})$; in particular, it is *not* adjacent to $p_{a_i}^{i,\alpha}$. As S only excludes edge-vertices in V_E that encode edges incident on $v_{a_i}^i$, it contains every vertex in $N(p_{a_i}^{i,\alpha}) \cap V_E$. Since S also contains every vertex in $K_{a_i}^{i,\alpha}$, we conclude that there is no path from $p_a^{i,\alpha}$ to $p_{n+2-a}^{i,\beta}$ in $H - S$.

Now, consider the case where $a_i < a$, i.e., $n+2-a < n+2-a_i$. In this case, it is convenient to consider a path from $p_{n+2-a}^{i,\beta}$ to $p_a^{i,\alpha}$. Once again, by construction of H, any path from $p_{n+2-a}^{i,\beta}$ to $p_a^{i,\alpha}$ in H contains a vertex of $K_{n+2-(a_i+1)}^{i,\beta} = K_{n+1-a_i}^{i,\beta}$ or of $N(p_{n+2-(a_i+1)}^{i,\beta}) \cap V_E = N(p_{n+1-a_i}^{i,\beta}) \cap V_E$. Recall that if edge $e \in E(G)$ is incident on $v_{a_i}^i$, then the corresponding edge-vertex v_e in H is adjacent only to vertices in $p^{i,\beta}[n-a_i+2, \cdots n+1]$ and $K^{i,\beta}[n-a_i+2, \cdots, n+1]$ in $V(B^{i,\beta})$; in particular, it is *not* adjacent to $p_{n+1-a_i}^{i,\beta}$. As S only excludes edge-vertices in V_E that encode edges incident on $v_{a_i}^i$, it contains every vertex in $N(p_{n+1-a_i}^{i,\beta}) \cap V_E$. Since S also contains every vertex in $K_{n+1-a_i}^{i,\beta}$, we conclude that there is no path from $p_{n+2-a}^{i,\beta}$ to $p_a^{i,\alpha}$. This implies that no terminal pair in P is connected in $H - S$ which concludes the proof. $\square$

Lemma 4.2 *If* (H, P, k) *is a* YES-*instance of* MULTICUT WITH UNDEL TERM*, then* $(G, q, (V_1, V_2, \ldots, V_q))$ *is a* YES-*instance of* MULTICOLORED CLIQUE.

Proof Assume that (H, P, k) is a YES-instance of MULTICUT WITH UNDEL TERM and let S be a P-multicut of H of size at most k. Recall that, by definition of the problem, $S \cap V(P) = \emptyset$. Also, recall that the reduction adds the clique K of size $2M^2$ and makes it complete to $\{p_{n+1}^{i,\alpha}, p_{n+1}^{i,\beta} \mid i \in [q]\}$ and V_E. Note that $K \setminus S \neq \emptyset$ as $k < 2M^2$.

Consider an index $i \in [q]$. It is easy to see that there exists $a \in [n]$ such that $K_a^{i,\alpha} \subseteq S$ as otherwise, there is a path from $p_1^{i,\alpha}$ to $p_{n+1}^{i,\beta}$ in $H - S$. Let $a_i \in [n]$ be the largest index such that $K_{a_i}^{i,\alpha} \subseteq S$. Similarly, there must exist $b \in [n]$ such that $K_b^{i,\beta} \subseteq S$: let $b_i \in [n]$ be the largest index such that $K_{b_i}^{i,\beta} \subseteq S$. Note that by definition of a_i, b_i and the fact that $K \setminus S \neq \emptyset$, there is path from $p_{a_i+1}^{i,\alpha}$ to $p_{b_i+1}^{i,\beta}$ in $H - S$. Now suppose for a contradiction that $a_i + 1 + b_i + 1 \leq n+2$. Then there exists $a_i' \in [n]$ such that $a_i' \geq a_i$ and $a_i' + 1 + b_i + 1 = n+2$ and so, by definition of P, $(p_{a_i'+1}^{i,\alpha}, p_{b_i+1}^{i,\beta}) \in P$. Moreover, by construction of H, the existence of a path from $p_{a_i+1}^{i,\alpha}$ to $p_{b_i+1}^{i,\beta}$ in $H - S$ implies that there is path from $p_{a_i'+1}^{i,\alpha}$ to $p_{b_i+1}^{i,\beta}$ in $H - S$; this however, contradicts the fact that S is a P-multicut of H. Therefore $a_i + 1 + b_i + 1 \geq n+3$, i.e., $a_i + b_i \geq n+1$. Since this holds for any $i \in [q]$, we have that

$$\left| S \cap \left(\bigcup_{i \in [q]} V(B^{i,\alpha}) \cup V(B^{i,\beta}) \right) \right| \geq \sum_{i \in [q]} (a_i M + b_i M) \geq q(n+1)M.$$

Since $|E(G)| - q(q-1)/2 < M$ and S has size at most $k = q(n+1)M + |E(G)| - q(q-1)/2$, it follows that, in fact, $a_i + b_i = n+1$ for all $i \in [q]$. Hence, $S \cap (\bigcup_{i \in [q]} V(B^{i,\alpha}) \cup V(B^{i,\beta}))$ corresponds to a collection of vertices $\{v^1_{a_1}, v^2_{a_2}, \ldots, v^q_{a_q}\}$ in G such that $v^i_{a_i} \in V_i$ for every $i \in [q]$.

In the remaining proof, we argue that there are at least $q(q-1)/2$ edges with endpoints in $\{v^1_{a_1}, v^2_{a_2}, \ldots, v^q_{a_q}\}$. Since $|E(G)| - q(q-1)/2 < M$, and every clique in $B^{i,\alpha}$ is of size at least M, for any $a \in [n]$ such that $a < a_i$, we have $K^{i,\alpha}_a \setminus S \neq \emptyset$. In other words, there is at least one vertex in $H - S$ from each clique $K^{i,\alpha}_a$ where $a < a_i$. Since a_i is the largest index such that $K^{i,\alpha}_{a_i} \subseteq S$, this also holds for every $a > a_i$. As S intersects every path from $p^{i,\alpha}_1$ to $p^{i,\beta}_{n+1}$, it contains every vertex in $N(p^{i,\alpha}_{a_i}) \cap V_E$. Using similar arguments, we conclude that S also contains every vertex in $N(p^{i,\beta}_{b_i}) \cap V_E = N(p^{i,\beta}_{n+1-a_i}) \cap V_E$. Now recall that if edge $e \in E(G)$ is incident on $v^i_{a_i}$, then the corresponding edge-vertex v_e in H is adjacent to

- Terminals in $V(B^{i,\alpha})$ which are in $p^{i,\alpha}[a_i + 1, n+1]$, and
- Terminals in $V(B^{i,\beta})$ which are in $p^{i,\beta}[n - a_i + 2, n+1]$.

In particular, v_e is *not* adjacent to $p^{i,\alpha}_{a_i}$ and $p^{i,\beta}_{n+1-a_i}$. This implies that only edges-vertices that correspond to edges incident on $v^i_{a_i}$ can be excluded from S. As this holds for any $i \in [q]$, every vertex in $V_E \setminus S$ has its endpoints in $\{v^1_{a_1}, v^2_{a_2}, \ldots, v^q_{a_q}\}$. As $|S \cap (\bigcup_{i \in [q]} V(B^{i,\alpha}) \cup V(B^{i,\beta}))| = q(n+1)M$ and $k = q(n+1)M + |E(G)| - q(q-1)/2$, we have $|S \cap V_E| \leq |E(G)| - q(q-1)/2$ which implies that $|V_E \setminus S| \geq q(q-1)/2$. Since G is a simple graph, it follows that $\{v^1_{a_1}, v^2_{a_2}, \ldots, v^q_{a_q}\}$ is a multicolored clique in G. This concludes the proof of the lemma. □

Finally, it is known that, assuming the ETH, there is no algorithm that can solve Multicolored Clique on instance $(G, q, (V_1, V_2, \ldots V_q))$ in time $f(q) \cdot |V(G)|^{o(q)}$ for any computable function f (see, e.g., [17, Corollary 14.23]). Thus, together with the fact that the reduction takes polynomial time in the size of the input, Lemma 4.1 and 4.2, and arguments that are standard for parameter preserving reductions, we conclude that the following holds.

Lemma 4.3 Multicut with Undeletable Terminals *on chordal graphs is* W[1]-hard *when parameterized by leafage ℓ and assuming the* ETH, *does not admit an algorithm running in time $f(\ell) \cdot n^{o(\ell)}$ for any computable function f.*

The remainder of this section is devoted to the proof of the following lemma, which together with Lemma 4.3 proves Theorem 1.2.

Lemma 4.4 Multicut with Undeletable Terminals *on chordal graph of leafage at most ℓ admits an* XP*-algorithm running in time $n^{\mathcal{O}(\ell)}$.*

Proof Let (G, P) be an instance of Multicut with UnDel Term where G is a chordal graph of leafage at most ℓ. Let $(T, \mathcal{M})$ be a tree representation of G of leafage at most ℓ. We say that a path in T is a *maximal degree-*2 *path* if it contains no branching nodes, except for possibly the first and last node of the path, and it cannot be extended without violating this property (that is, it is maximal). A P-multicut S of G is said to *destroy* an edge $e \in E(T)$ if $\texttt{ver}(e) \subseteq S$.

Let us root T at an arbitrary node $\mathsf{r} \in V(T)$. Since the number of leaves of T is at most ℓ, T has at most $2\ell - 2$ maximal degree-2 paths, one starting at each each leaf or branching node (except the root) and ending at the first ancestor in T which is a branching node.

Now for each maximal degree-2 path Q from α to β in T, guess the first (i.e., closest to α) and last (i.e., closest to β) edge of Q, say e_1^Q and e_2^Q, respectively, such that S destroys e_1^Q and e_2^Q. Note that, it might be the case that an optimal solution does not destroy an edge of Q or only destroys one edge of Q (i.e., $e_1^Q = e_2^Q$). Since the length of any maximal degree-2 path is $\mathcal{O}(n)$, this creates at most $(n+1)^{2\ell}$ branches.

In each such branch, let $D \subseteq E(T)$ be the set of guessed edges of T. Pick $V_D = \{\texttt{ver}(e) \mid e \in D\}$ in the solution and delete V_D from G: let (G', P') be the resulting instances and further let T' be obtained from T by deleting the edges in D and set $\mathcal{M}' = \mathcal{M}|_{V(G')}$. Observe that the tree representation of each connected component of G' is given by some tree of the forest T' together with $\mathcal{M}'$ restricted to the vertices of the corresponding connected component. Note that it is enough to solve the problem independently on each connected component of G'.

Thus, without loss of generality, assume that G' is connected and let $(T', \mathcal{M}')$ be a tree representation of G' as defined above. Suppose that G' has at least one terminal pair in P', say $(s, t) \in P' \subseteq P$. If T' is a path, i.e., G' is an interval graph, then the problem can be solved in polynomial time [29, Theorem 5]. Otherwise, we ignore this branch.

The algorithm outputs a solution if there is at least one branch where a solution was computed. Otherwise, there is no solution.

It is not difficult to see that the above algorithm indeed solves the problem, as it considers all the possible ways a solution could intersect every maximal degree-2 path. □

5 Multiway Cut with Undeletable Terminals on Chordal Graphs

In this section, we consider the Multiway Cut with Undeletable Terminals problem formally defined below. Given a graph G and a set $P \subseteq V(G)$, a set $S \subseteq V(G) \setminus P$ is a called a P-*multiway-cut* in G if $G - S$ has no (p, p')-path for any two distinct $p, p' \in P$.

Multiway Cut with Undeletable Terminals (MWC)
Input: An undirected graph G and a set $P \subseteq V(G)$ of terminals.
Question: Find the size of a minimum P-multiway-cut in G.

The aim of this section is to prove Theorem 1.3 which states that Multiway Cut with Undeletable Terminals can be solved in $n^{\mathcal{O}(1)}$-time on chordal graphs. Before turning to the proof, we first start with a few definitions. Let $(T, \mathcal{M})$ a tree representation of a chordal graph G where T is rooted at an arbitrary node $\mathsf{r} \in V(T)$. Given a subtree T' of T and a set $Q \subseteq V(G)$, we let $Q_{|T'} \subseteq Q$ be the set of vertices $x \in Q$ such that $\mathcal{M}(x) \subseteq V(T')$. Now let $Q \subseteq V(G)$ be an independent set of G such that for every leaf η of T, $\mathtt{ver}(\eta) \cap Q \neq \emptyset$. Then the *truncated tree w.r.t.* Q is the tree T_Q^{trunc} obtained from T as follows. Let $\{\eta_1, \ldots, \eta_q\}$ be the set of leaves of T. For each $i \in [q]$, let $Q_i \subseteq Q \setminus \mathtt{ver}(\mathsf{r})$ be the set of vertices $p \in Q \setminus \mathtt{ver}(\mathsf{r})$ such that $\mathtt{top}_{\mathcal{M}}(p)$ is on the (η_i, r)-path in T, and let $p_i \in Q_i$ be the vertex of Q_i such that $\mathtt{top}_{\mathcal{M}}(p_i)$ is closest to r. Then T_Q^{trunc} is obtained from T by deleting the subtrees rooted at the children of the nodes in $\{\mathtt{top}_{\mathcal{M}}(p_i) \mid i \in [q]\}$. Note that, by construction, the set of leaves of T_Q^{trunc} is $\{\mathtt{top}_{\mathcal{M}}(p_i) \mid i \in [q]\}$ and that, apart from the vertices in $\{p_i \mid i \in [q]\}$, there is at most one other vertex in Q whose model intersects $V(T_Q^{\mathsf{trunc}})$, namely the potential vertex in $Q \cap \mathtt{ver}(\mathsf{r})$ (note that if such a vertex exists, its model is in fact fully contained in T_Q^{trunc}). Finally, given a set $P \subseteq V(G)$, a P-multiway-cut X in G is said to *destroy* an edge $e \in E(T)$ if $\mathtt{ver}(e) \subseteq X$.

We now turn to the proof of Theorem 1.3. Throughout the remaining of this section, we let (G, P) be an instance of MWC, where G is a n-vertex chordal graph, and further let $(T, \mathcal{M})$ be a tree representation of G. First, we may assume that P is an independent set: indeed, if there exist $p, p' \in P$ such that $pp' \in E(G)$, then (G, P) is a No-instance. Furthermore, if a vertex $v \in V(G)$ does not belong to any (p, p')-path in G, where $p, p' \in P$, then it can be safely deleted as no minimal P-multiway-cut in G may contain v. Hence, we assume that every vertex in G participates in some (p, p')-path where $p, p' \in P$; in particular, we may assume that for every leaf η of T, $\mathtt{ver}(\eta) \cap P \neq \emptyset$. Note that, consequently, for every internal node $\alpha \in V(T)$, the truncation of T_α w.r.t. $P_{|T_\alpha}$ exists.

Now let T_0 be the tree obtained by adding a new node r_0 and connecting it to an arbitrary node $\mathsf{r} \in V(T)$. Observe that $(T_0, \mathcal{M})$ is also a tree representation of G. In the following, we root T_0 at r_0. To prove Theorem 1.3, we design a dynamic program that computes, in a bottom-up traversal of T_0, the entries of a table $\mathtt{A}$ whose content is defined as follows. The table $\mathtt{A}$ is indexed over the edges of $E(T_0)$. For each node $\alpha \in V(T)$, $\mathtt{A}[\alpha\mathtt{parent}_{T_0}(\alpha)]$ stores the size of a minimum $P_{|T_\alpha}$-multiway-cut in $G_{|T_\alpha}$. The size of a minimum P-multiway-cut in G may then be found in $\mathtt{A}[\mathsf{r}\mathsf{r}_0]$. We describe below how to compute the entries of $\mathtt{A}$.

Update Procedure For every leaf η of T, we set $\mathtt{A}[\eta\mathtt{parent}_{T_0}(\eta)] = 0$. Consider now an internal node α of T. We show how to compute $\mathtt{A}[\alpha\mathtt{parent}_{T_0}(\alpha)]$ assuming that for every edge $e \in E(T_\alpha)$, the entry $\mathtt{A}[e]$ is correctly filled.

Let $\widetilde{T}$ be the truncation of T_α w.r.t. $P_{|T_\alpha}$ and let $\widetilde{G} = G_{|\widetilde{T}}$. Denote by $\eta_1, \ldots, \eta_q$ the leaves of $\widetilde{T}$. Recall that, by construction, for every $i \in [q]$, there exists $p_i \in P_{|T_\alpha}$ such that $\eta_i = \mathtt{top}_{\mathcal{M}}(p_i)$: we let $\widetilde{P} = \{p_i \mid i \in [q]\}$. Furthermore, it may be that $P_{|T_\alpha} \cap \mathtt{ver}(\mathsf{r})$ is nonempty: we let $\widetilde{P}_{\mathsf{r}} = P_{|T_\alpha} \cap \mathtt{ver}(\mathsf{r})$. Note that $|\widetilde{P}_{\mathsf{r}}| \leq 1$: if $\widetilde{P}_{\mathsf{r}} \neq \emptyset$ then we refer to the terminal in $\widetilde{P}_{\mathsf{r}}$ as the *root terminal*. Observe that $V(\widetilde{G}) \cap P_{|T_\alpha} = V(\widetilde{G}) \cap P = \widetilde{P} \cup \widetilde{P}_{\mathsf{r}}$ by construction. To compute $\mathtt{A}[\alpha\mathtt{parent}_{T_0}(\alpha)]$, we distinguish two cases:

(1) if $\widetilde{P}_{\mathsf{r}} \neq \emptyset$ then we construct a unique instance $(H_0, \mathtt{s}, \mathtt{t}, \mathtt{wt}_0)$ of (s,t)- Cut;
(2) Otherwise, for every $i \in [0,q]$, we construct an instance $(H_i, \mathtt{s}, \mathtt{t}, \mathtt{wt}_i)$ of (s,t)- Cut.

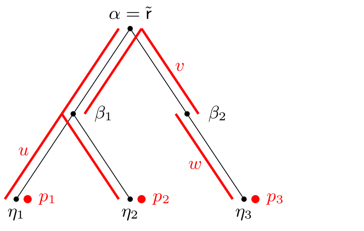
We describe below how such instances are constructed. First, recall that an instance of the (s,t)- Cut problem consists of a digraph D, vertices $s, t \in V(D)$, a weight function $\mathtt{wt} : E(D) \to \mathbb{N} \cup \{\infty\}$, and the goal is to find a set $X \subseteq E(D)$ such that $D - X$ has no (s,t)-path and $\mathtt{wt}(X)$ is minimum with this property, where $\mathtt{wt}(X) = \sum_{u \in X} \mathtt{wt}(u)$.

Construction of the (s,t)- Cut *Instances* For every $i \in [q]$, let us denote by $\widetilde{P}_i = \widetilde{P} \setminus \{p_i\}$ and let $\widetilde{P}_0 = \widetilde{P}$. Consider $i \in [0,q]$. Before turning to the formal construction of the instance $(H_i, \mathtt{s}, \mathtt{t}, \mathtt{wt}_i)$, let us first give an intuitive idea of the construction. The digraph H_i is obtained from $\widetilde{T}$ by orienting all edges of $\widetilde{T}$ towards its root $\tilde{\mathsf{r}} = \alpha$ and further adding vertices and weighted arcs to encode the graph $G_{|T_\alpha}$. The arcs in H_i corresponding to the edges of $\widetilde{T}$ are called the *tree arcs* and the nodes in H_i corresponding to the nodes of $\widetilde{T}$ are called the *tree nodes*. The idea is that we separate, for each terminal $p \in \widetilde{P}_i$, the node $\mathtt{top}_{\mathcal{M}}(p)$ from the root $\tilde{\mathsf{r}}$. To achieve this, we add a *source* node $\mathtt{s}$ and *source* arcs from $\mathtt{s}$ to $\mathtt{top}_{\mathcal{M}}(p)$ (of infinite weight) and look for an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut in H_i. Since the edges of T can presumably not be independently destroyed in a P-multiway-cut, we need some additional vertices to encode these dependencies. For each vertex $v \in V(\widetilde{G}) \setminus \widetilde{P}_i$, we introduce a node $\gamma(v)$ in H_i which is reachable via *connection* arcs (with infinite weight) from all the tree nodes that are contained in the model of v. This node $\gamma(v)$ is further connected via a *sink* arc (of weight one) to $\mathtt{top}_{\mathcal{M}}(v)$ which ensures that if we want to cut a tree arc, we also have to cut all the sink arcs associated to vertices containing the corresponding edge in their model. The index i is then used to specify which root-to-leaf path of $\widetilde{T}$ is uncut: if $i = 0$ then every such path is cut, otherwise the $(\eta_i, \tilde{\mathsf{r}})$-path is uncut. To encode the rest of the solution, we associate with each tree arc (β, δ) a weight $\mathtt{wt}_i((\beta,\delta))$ corresponding to the size of a minimum $P_{|\beta}$-multiway-cut in $G_{|\beta}$.

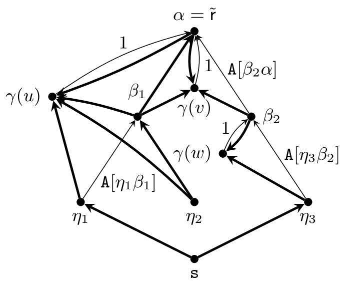
We proceed with the formal construction of H_i. The vertex set of H_i is $V(H_i) = V(\widetilde{T}) \uplus \{\mathtt{s}\} \uplus \{\Gamma\}$ where $\Gamma = \{\gamma(v) \mid v \in V(\widetilde{G}) \setminus \widetilde{P}\}$, that is, Γ contains a node of every non-terminal vertex in $\widetilde{G}$. For every $z \in \Gamma$, we denote by $\gamma^{-1}(z)$ the corresponding vertex in $V(\widetilde{G}) \setminus \widetilde{P}$. The arc set of H_i is partitioned into four sets:

- The set $E_{\widetilde{T}}$ of *tree arcs* containing all the edges of $\widetilde{T}$ oriented towards the root $\tilde{\mathsf{r}}$,
- The set $E^i_{\mathtt{source}} = \{(\mathtt{s}, \mathtt{top}_{\mathcal{M}}(p)) \mid p \in \widetilde{P}_i\}$ of *source* arcs,
- The set $E_{\mathtt{conn}} = \{(\alpha, \gamma(v)) \mid \gamma(v) \in \Gamma, \alpha \in \mathcal{M}(v) \cap V(\widetilde{T})\}$ of *connection* arcs and
- The set $E_{\mathtt{sink}} = \{(\gamma(v), \mathtt{top}_{\mathcal{M}}(v)) \mid v \in V(\widetilde{G}) \setminus \widetilde{P}\}$ of *sink* arcs.

Furthermore, if $\widetilde{P}_{\mathsf{r}} \neq \emptyset$, then we let $E_{\mathtt{rterm}} \subseteq E_{\widetilde{T}}$ be the set of tree arcs $(\beta, \delta) \in E_{\widetilde{T}}$ such that the edge $\beta\delta$ is contained in the model of the root terminal; otherwise, we let $E_{\mathtt{rterm}} = \emptyset$. The weight function $\mathtt{wt}_i : E(H_i) \to \mathbb{N} \cup \{\infty\}$ is defined as follows. For every $j \in [q]$, let ρ_j be the path in $\widetilde{T}$ from η_j to $\tilde{\mathsf{r}}$ and let $\overrightarrow{\rho_j}$ be the corresponding directed path in H_i (that is, $\overrightarrow{\rho_j}$ is the path in H_i from η_j to $\tilde{\mathsf{r}}$ consisting only of tree arcs). Then for every arc e of H_i,



(a) The tree representation $(\widetilde{T}, \mathcal{M}_{|V(\widetilde{G})})$ of $\widetilde{G}$ where $V(\widetilde{G}) = \{p_1, p_2, p_3, u, v, w\}$ and $\widetilde{P}_{\mathsf{r}} = \emptyset$.


$\alpha = \tilde{\mathsf{r}}$
1
1
$\mathtt{A}[\beta_2\alpha]$
$\gamma(u)$
β_1
$\gamma(v)$
β_2
1
$\gamma(w)$
$\mathtt{A}[\eta_3\beta_2]$
$\mathtt{A}[\eta_1\beta_1]$
η_1
η_2
η_3
s

(b) The instance $(H_2, \mathtt{s}, \tilde{\mathsf{r}}, \mathtt{wt}_2)$ (thick arcs have infinite weight).

Fig. 3 An illustration of the construction of the (s, t)- Cut instances

$$\mathtt{wt}_i(e) = \begin{cases} \mathtt{A}[e] & \text{if } i = 0 \text{ and } e \in E_{\widetilde{T}} \setminus E_{\mathtt{rterm}} \\ \mathtt{A}[e] & \text{if } i \neq 0,\ e \in E_{\widetilde{T}} \text{ and } e \text{ does not belong to the path } \overrightarrow{\rho_i} \\ 1 & \text{if } e \in E_{\mathtt{sink}} \\ \infty & \text{otherwise.} \end{cases}$$

Note, in particular, that every arc in $E_{\mathtt{rterm}}$ (if any) has infinite weight. Similarly, if $i \neq 0$, then every arc of the path $\overrightarrow{\rho_i}$ has infinite weight. This completes the construction of the instance $(H_i, \mathtt{s}, \mathtt{t} = \tilde{\mathsf{r}}, \mathtt{wt}_i)$ (see Fig. 3). It is easy to see that such an instance can be constructed in $\mathcal{O}(n^2)$-time.

Now let X_0 be an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut in H_0 such that $\mathtt{wt}_0(X_0)$ is minimum; and if $\widetilde{P}_{\mathsf{r}} = \emptyset$, then for every $i \in [q]$, further let X_i be an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut in H_i such that $\mathtt{wt}_i(X_i)$ is minimum. For each $i \in [q]$, let us denote by $\mathtt{cost}_i = \mathtt{A}[\eta_i \mathtt{parent}_{T_0}(\eta_i)]$ and let $\mathtt{cost}_0 = 0$. Then we set

$$\mathtt{A}[\alpha \mathtt{parent}_{T_0}(\alpha)] = \begin{cases} |X_0| & \text{if } \widetilde{P}_{\mathsf{r}} \neq \emptyset \\ \min_{i \in [0,q]}\{|X_i| + \mathtt{cost}_i\} & \text{otherwise} \end{cases}$$

In the following, for convenience, we let $I = [0, q]$ if $\widetilde{P}_{\mathsf{r}} = \emptyset$, and $I = \{0\}$ otherwise. We next show that the entry $\mathtt{A}[\alpha \mathtt{parent}_{T_0}(\alpha)]$ is updated correctly. To this end, we show that $G_{|T_\alpha}$ has a $P_{|T_\alpha}$-multiway-cut of size at most k if and only if there exists $i \in I$ such that H_i has an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut of weight at most $k - \mathtt{cost}_i$ w.r.t. $\mathtt{wt}_i$.

Lemma 5.1 *For any $i \in I$, if H_i has an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut Y such that $\mathtt{wt}_i(Y) \leq k - \mathtt{cost}_i$, then $G_{|T_\alpha}$ has a $P_{|T_\alpha}$-multiway-cut of size at most k.*

Proof Assume that there exists $i \in I$ such that H_i has an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut Y where $\mathtt{wt}_i(Y) \leq k - \mathtt{cost}_i$. For every $j \in [q] \setminus \{i\}$, let A_j be the set of tree arcs on the path $\overrightarrow{\rho_j}$ belonging to Y (recall that $\overrightarrow{\rho_j}$ is the path in H_i from η_j to $\tilde{\mathsf{r}}$ consisting only of tree arcs). Note that since Y is an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut, $A_j \neq \emptyset$ for every $j \in [q] \setminus \{i\}$.

Claim 5.2 *For every terminal $j \in [q] \setminus \{i\}$, there exists an arc $(x, y) \in A_j$ such that for every $z \in N^+_{H_i}(x) \setminus (N^-_{H_i}(x) \cup \{y\})$, the sink arc with tail z belongs to Y.*

Proof Suppose for a contradiction that this does not hold for some index $j \in [q] \setminus \{i\}$, that is, for every arc $(x, y) \in A_j$, there exists $z \in N^+_{H_i}(x) \setminus (N^-_{H_i}(x) \cup \{y\})$ such that the sink arc with tail z does not belong to Y. Let $(x_1, y_1), \ldots, (x_a, y_a)$ be the arcs of A_j ordered according to their order of appearance when traversing the path $\overrightarrow{\rho_j}$. We show that, in this case, there is a path from $\mathtt{s}$ to $\tilde{\mathtt{r}}$ in $H - Y$. For every $b \in [a]$, denote by $Z_b \subseteq N^+_{H_i}(x_b) \setminus (N^-_{H_i}(x_b) \cup \{y_b\})$ the set of vertices z such that the sink arc with tail z does not belong to Y. Let $b_1, \ldots, b_w \in [a]$ be the longest sequence defined as follows:

- $b_1 \in [a]$ is the largest index such that $Z_1 \cap Z_{b_1} \neq \emptyset$ and
- For every $l > 1$, $b_l \in [a]$ is the largest index such that $Z_{b_{l-1}+1} \cap Z_{b_l} \neq \emptyset$.

For every $l \in [w]$, consider a vertex $z_{b_l} \in Z_{j_l}$ and let $h_{b_l} \in N^+_{H_i}(z_{b_l})$ be the head of the sink arc with tail z_{b_l}. Then for every $l \in [w-1]$, h_{b_l} lies on the path $\overrightarrow{\rho_j}[y_{b_l}, x_{b_l+1}]$: indeed, since $z_{b_l} \notin Z_{b_l+1}$ by the choice of b_l, either $z_{b_l} \notin N^+_{H_i}(x_{b_l+1})$ or $z_{b_l} \in N^+_{H_i}(x_{b_l+1}) \cap N^-_H(x_{b_l+1})$; but $z_{b_l} \in N^+_{H_i}(x_{b_l}) \setminus N^-_{H_i}(x_{b_l})$ by construction, and so, h_{b_l} necessarily lies on $\overrightarrow{\rho_j}[y_{b_l}, x_{b_l+1}]$.

Now observe that, by maximality of the sequence, $b_w = a$: indeed, if $b_w < a$ then the sequence could be extended as $Z_{b_w+1} \neq \emptyset$ by assumption. Since $z_{b_w} \notin N^-_{H_i}(x_{b_w})$, this implies, in particular, that h_{b_w} lies on the path $\overrightarrow{\rho_j}[y_{b_w}, \tilde{\mathtt{r}}]$. It follows that

$$\mathtt{s}\overrightarrow{\rho_j}[\eta_j, x_1]z_{b_1}\overrightarrow{\rho_j}[h_{b_1}, x_{b_1+1}]z_{b_2} \ldots z_{b_l}\overrightarrow{\rho_j}[h_{b_l}, x_{b_l+1}]z_{b_{l+1}} \ldots \overrightarrow{\rho_j}$$
$$[h_{b_{w-1}}, x_{b_{w-1}+1}]z_{b_w}L[h_{b_w}, \tilde{\mathtt{r}}]$$

is a path from $\mathtt{s}$ to $\tilde{\mathtt{r}}$ in $H - Y$, a contradiction which proves our claim.

For every $j \in [q] \setminus \{i\}$, let $e_j = (x_j, y_j) \in A_j$ be the arc closest to $\tilde{\mathtt{r}}$ such that for every $z \in N^+_{H_i}(x_j) \setminus (N^-_{H_i}(x_j) \cup \{y_j\})$, the sink arc with tail z belongs to Y (note that we may have $e_j = e_{j'}$ for two distinct $j, j' \in [q] \setminus \{i\}$). Denote by $E = \{e_j \mid j \in [q] \setminus \{i\}\} \cup \{e^*\}$ where $e^* = (\eta_i, \mathtt{parent}(\eta_i))$. For every $e = (x, y) \in E$, let $\widetilde{P}_e \subseteq \widetilde{P}_i$ be the set of terminals in $\widetilde{P}_i$ which are also terminals in the instance restricted to T_x. Note that $\{P_e \mid e \in E \setminus \{e^*\}\}$ is a partition of $\widetilde{P}_i$: indeed, by construction, every $p \in \widetilde{P}_i$ belongs to at least one such set and if there exist $e, e' \in E \setminus \{e^*\}$ such that $\widetilde{P}_e \cap \widetilde{P}_{e'} \neq \emptyset$, then for any $j \in [q] \setminus \{i\}$ such that $p_j \in P_e \cap P_{e'}$, $e, e' \in A_j$; in particular, both e and e' lie on the path $\overrightarrow{\rho_j}$, a contradiction to the choice of the arc in A_j.

Now for every $e = (x, y) \in E$, let S_e be a minimum $P_{|T_x}$-multiway-cut in $G_{|T_x}$ and denote by $N_e = N^+_{H_i}(x) \setminus (N^-_{H_i}(x) \cup \{y\})$. We define

$$S = S_{e^*} \cup \bigcup_{e \in E \setminus \{e^*\}} S_e \cup \{\gamma^{-1}(z) \mid z \in N_e\}.$$

Claim 5.3 *S is a $P_{|T_\alpha}$-multiway-cut in $G_{|T_\alpha}$.*

Proof Since for every $e = (x, y) \in E$, S_e is a $\widetilde{P}_{|T_x}$-multiway-cut in $G_{|T_x}$, it is in fact enough to show that for every $e, e' \in E$, $p \in \widetilde{P}_e$ and $p' \in \widetilde{P}_{e'}$, there is no path from p to p' in $G_{|T_\alpha} - S$.

Consider therefore $j, j' \in [q] \setminus \{i\}$ such that $p_j \in \widetilde{P}_e$ and $p_{j'} \in \widetilde{P}_{e'}$ for two distinct $e, e' \in E$. Since, as shown above, $\{\widetilde{P}_f \mid f \in E \setminus \{e^*\}\}$ is a partition of $\widetilde{P}_i$, $p_{j'} \notin \widetilde{P}_e$ and $p_j \notin \widetilde{P}_{e'}$; in particular, e' does not lie on the path $\overrightarrow{\rho_j}$ and e does not lie on the path $\overrightarrow{\rho_{j'}}$. It follows that any path in $G_{|T_\alpha}$ from p_j to $p_{j'}$ contains at least one vertex x whose model contains the edge corresponding to e; but then, $\gamma(x) \in N_e$ and so, $x \in S$ by construction. Thus, there is no path from p_j to $p_{j'}$ in $G_{|T_\alpha} - S$.

Finally, note that, by construction,

$$\begin{aligned}
|S| &= |S_{e^*}| + \sum_{e \in E \setminus \{e^*\}} |S_e| + |*| \bigcup_{e \in E \setminus \{e^*\}} \{\gamma^{-1}(z) \mid z \in N_e\} \\
&= |S_{e^*}| + \sum_{e \in E \setminus \{e^*\}} \mathtt{wt}_i(e) + \sum_{z \in \bigcup_{e \in E \setminus \{e^*\}} N_e} \mathtt{wt}_i((z, \mathtt{top}_{\mathcal{M}}(\gamma^{-1}(z)))) \\
&\leq \mathtt{cost}_i + \mathtt{wt}_i(Y) \leq k
\end{aligned}$$

which concludes the proof. $\square$

Lemma 5.4 *If $G_{|T_\alpha}$ has a $P_{|T_\alpha}$-multiway-cut X of size at most k, then there exists $i \in I$ such that H_i has an $(\mathtt{s}, \tilde{\mathtt{r}})$-cut Y where $\mathtt{wt}_i(Y) \leq k - \mathtt{cost}_i$.*

Proof Recall that for every $j \in [q]$, ρ_j is the unique $(\eta_j, \tilde{\mathtt{r}})$-path in $\widetilde{T}$. To prove the lemma, we first show the following.

Claim 5.5 *If there exists $i \in [q]$ such that $G_{|T_\alpha}$ has a $P_{|T_\alpha}$-multiway-cut X of size at most k where*

(1) X does not destroy any edge of ρ_i and
(2) For every $j \in [q] \setminus \{i\}$, X destroys an edge of ρ_j,

then H_i has an $(\mathtt{s}, \tilde{\mathtt{r}})$-cut Y such that $\mathtt{wt}_i(Y) \leq k - \mathtt{cost}_i$.

Proof Assume that such an index $i \in [q]$ exists and let X be a $P_{|T_\alpha}$-multiway-cut X of size at most k satisfying item (1) and (2). Note that since X does not destroy any edge of ρ_i, $\widetilde{P}_{\mathtt{r}} = \emptyset$ for, otherwise, p_i and the root terminal would be in the same connected component of $G_{|T_\alpha} - X$ thereby contradicting the fact that X is a $P_{|T_\alpha}$-multiway-cut. For every $j \in [q] \setminus \{i\}$, let $e_j \in E(\widetilde{T})$ be the closest edge to η_j on ρ_j such that $\mathtt{ver}(e_j) \subseteq X$ (note that the edges $e_1, \ldots, e_q$ are not necessarily pairwise distinct). Denote by $E = \{e_j \mid j \in [q] \setminus \{i\}\}$. We construct an $(\mathtt{s}, \tilde{\mathtt{r}})$-cut Y in H_i as follows: Y contains the tree arcs of H_i corresponding to the edges in E and for each $v \in X$ such that $\mathcal{M}(v)$ contains at least one edge of E (that is, $v \in \mathtt{ver}(e)$ for some edge $e \in E$), we include in Y the sink arc $(\gamma(v), \mathtt{top}_{\mathcal{M}}(v))$ of $E(H_i)$. Let us show that Y is indeed an $(\mathtt{s}, \tilde{\mathtt{r}})$-cut in H_i.

For every $j \in [q] \setminus \{i\}$, let $V_-^j \subseteq V(\widetilde{T})$ ($V_+^j \subseteq V(\widetilde{T})$, respectively) be the set of nodes of the subpath of ρ_j from η_j to the tail of e_j (the head of e_j to $\tilde{\mathtt{r}}$, respectively). We contend that for every $j \in [q] \setminus \{i\}$, there is no (V_-^j, V_+^j)-path in $H_i - Y$. Note that if true, this would prove that Y is indeed an $(\mathtt{s}, \tilde{\mathtt{r}})$-cut in H_i. For the sake of contradiction, suppose that, for some $j \in [q] \setminus \{i\}$, there is a path L in $H_i - Y$ from a vertex $x \in V_-^j$ to a vertex $y \in V_+^j$. Since the tree arc in H_i corresponding e_j belongs to Y, there must exist a vertex $z \in V(L)$ such that $N_{H_i}^-(z) \cap V_-^j \cap V(L) \neq \emptyset$ and $N_{H_i}^+(z) \cap V_+^j \cap V(L) \neq \emptyset$; in particular, the sink arc e with tail z must belong to L. By construction of H_i, it must then be that $\mathcal{M}(\gamma^{-1}(z))$ contains the edge e_j, that is, $\gamma^{-1}(z) \in \mathtt{ver}(e_j)$; but then, $\gamma^{-1}(z) \in X$ and so, $e \in Y$ by construction, a contradiction which proves our claim.

Let us finally show that $\mathtt{wt}_i(Y) \leq k - \mathtt{cost}_i$. To this end, for every $e \in E$, let $X_e \subseteq X$ be the restriction of X to T_{t_e} where t_e is the endpoint of e the furthest from $\tilde{\mathsf{r}}$ (note that for any two distinct $e, e' \in E$, $X_e \cap X_{e'} = \emptyset$). Then, for every $e \in E$, X_e is a $P_{|T_{t_e}}$-multiway-cut in $G_{|T_{t_e}}$ and so, $\mathtt{wt}_i(e) \leq |X_e|$. Similarly, the restriction X_i of X to T_{η_i} is a $P_{|T_{\eta_i}}$-multiway-cut in $G_{|T_{\eta_i}}$ and so, $|X_i| \geq \mathtt{cost}_i$ (note that, by construction, $X_i \cap X_e = \emptyset$ for every $e \in E$). Letting $X' = \bigcup_{e \in E} \mathtt{ver}(e)$, it then follows from the definition of Y that

$$\mathtt{wt}_i(Y) = |X'| + \sum_{e \in E} \mathtt{wt}_i(e) \leq |X'| + \sum_{e \in E} |X_e| \leq |X| - |X_i| \leq k - \mathtt{cost}_i$$

as $X' \cap X_i = \emptyset$ and for every $e \in E$, $X' \cap X_e = \emptyset$.

Using similar arguments, we can also prove the following.

Claim 5.6 *If $G_{|T_\alpha}$ has a $P_{|T_\alpha}$-multiway-cut X of size at most k such that for every $i \in [q]$, X destroys an edge of ρ_i, then H_0 has an $(\mathtt{s}, \tilde{\mathsf{r}})$-cut Y such that $\mathtt{wt}_i(Y) \leq k$.*

To conclude the proof of Lemma 5.4, let us show that for any $P_{|T_\alpha}$-multiway-cut S in $G_{|T_\alpha}$, S destroys an edge of every root-to-leaf path of $\widetilde{T}$, except for at most one when $\widetilde{P}_{\mathsf{r}} = \emptyset$. Note that if the claim is true, the lemma would then follow from Claims 5.5 and 5.6.

Let S be a $P_{|T_\alpha}$-multiway-cut in $G_{|T_\alpha}$. Observe first that if $\widetilde{P}_{\mathsf{r}} \neq \emptyset$ then for every $i \in [q]$, S must destroy an edge of ρ_i for, otherwise, p_i and the root terminal are in the same connected component of $G_{|T_\alpha} - S$, thereby contradicting the fact that S is a $P_{|T_\alpha}$-multiway-cut. Assume therefore that $\widetilde{P}_{\mathsf{r}} = \emptyset$ and suppose, for the sake of contradiction, that there exist two distinct indices $i, j \in [q]$ such that S destroys no edge of ρ_i and no edge of ρ_j. Then for every edge e of $\rho_i \cup \rho_j$, $\mathtt{ver}(e) \setminus S \neq \emptyset$: for each such edge e, let $\alpha_e \in \mathtt{ver}(e) \setminus S$. It is now not difficult to see that there is a path in $G_{|T_\alpha} - S$ from p_i to p_j using only vertices from $\{\alpha_e \mid e \text{ is an edge of } \rho_i \cup \rho_j\}$, a contradiction to the fact that S be a $P_{|T_\alpha}$-multiway-cut in $G_{|T_\alpha}$. □

We now conclude by Lemmas 5.1 and 5.4, that $\mathtt{A}[\alpha\mathtt{parent}_{T_0}(\alpha)]$ indeed stores the size of a minimum $P_{|T_\alpha}$-multiway-cut in $G_{|T_\alpha}$. Since the construction of each H_i takes polynomial-time, an $(\mathtt{s}, \mathtt{t})$-cut in H_i can be computed in polynomial time (see, for instance, [24]) and the number of H_is is at most n, it takes plynomial-time to update $\mathtt{A}[\alpha\mathtt{parent}_{T_0}(\alpha)]$. Finally, since the number of edges of T is linear in n, the overall running time is polynomial in n, which proves Theorem 1.3. We remark that a more careful analysis of the running time of the algorithm leads to an upper bound of $\mathcal{O}(n^4)$.

6 Restricting to H_ℓ-Induced-Subgraph-Free Chordal Graphs

In this section, we consider problems restricted to H_ℓ-induced-subgraph-free chordal graphs. Recall that H_ℓ is the split graph on 2ℓ vertices such that if $V(H_\ell) = C \uplus I$ is a split partition then (i) $|C| = |I| = \ell$, (ii) every vertex in C is adjacent to exactly one vertex in I, and (iii) every vertex in I is adjacent to exactly one vertex in C. As mentioned in the Introduction, the class of H_ℓ-induced-subgraph-free chordal graphs is a natural generalization of the class of chordal graphs of leafage at most ℓ. In fact, denoting by $\mathcal{C}_\ell$ the collection of all chordal graphs that have leafage at most ℓ and by $\mathcal{C}^{is}_\ell$ the collection of all chordal graphs that do not contain H_ℓ as a induced subgraph (that is, the collection of H_ℓ-induced-subgraph-free chordal graphs), the following holds.

Observation 6.1 $\mathcal{C}_\ell \subsetneq \mathcal{C}^{is}_{\ell+1}$.

Let us briefly explain why Observation 6.1 holds true. Walter generalized the concept of asteroidal triple in order to characterize other subclasses of chordal graphs [48] as follows. A subset of nonadjacent vertices of G is an *asteroidal set* if the removal of the closed neighborhood of any one of its elements does not disconnect the remaining ones. Formally, a set of vertices A of a graph G is *asteroidal* if for each $a \in A$, the vertices in $A\backslash\{a\}$ belong to a common connected component of $G - N[a]$. The asteroidal number of G, denoted by $\texttt{at}(G)$, is then the size of a largest asteroidal set of G. Note that in the graph $H_{\ell+1}$, I is an asteroidal set of size $\ell + 1$ and thus, $\texttt{at}(H_{\ell+1}) \geq \ell + 1$. By definition, if H is a subgraph of G and H is connected, then $\texttt{at}(H) \leq \texttt{at}(G)$. Lin et al. [40, Therorem 1] proved that for a connected chordal graph G, $\texttt{at}(G) \leq \texttt{lf}(G)$. Hence, if $\texttt{lf}(G) \leq \ell$, then it cannot contain $H_{\ell+1}$ as an induced subgraph. This implies that $\mathcal{C}_\ell \subseteq \mathcal{C}^{is}_{\ell+1}$. To see that $\mathcal{C}_\ell$ is proper subset of $\mathcal{C}^{is}_{\ell+1}$, consider a graph obtained from a star by subdividing every edge once. Then it is easy to see that this graph does not contain H_3 as induced subgraph but can have unbounded leafage.

The remainder of this section is organized as follows. In Sect. 6.1, we argue that the FPT algorithms for domination problems cannot be generalized to this larger graph class. We complement this with an XP-algorithm, which is optimal under the ETH. In Sect. 6.2, we present a simple algorithm to prove that MULTICUT WITH UNDEL TERM is paraNP-hard on this graph class. This implies that the XP-algorithm presented in Sect. 4 cannot be generalized for this larger class.

6.1 Dominating Set and Related Problems

In this subsection, we prove Theorem 1.4. We first show the hardness results of the theorem and provide afterwards the XP-algorithms for the problems.

Lemma 6.2 DOMINATING SET, CONNECTED DOMINATING SET *and* STEINER TREE *on H_ℓ-induced-subgraph-free chordal graphs are* W[1]-hard *when parameterized by ℓ and assuming the* ETH, *do not admit an algorithm running in time $f(\ell) \cdot n^{o(\ell)}$ for any computable function f.*

Proof We present a parameter preserving reduction from Multicolored Independent Set. An instance of this problem consists of a graph G, an integer q, and a partition $(V_1, \dots, V_q)$ of $V(G)$. The objective is to determine whether G has an independent set which contains exactly one vertex from every part V_i. We assume, without loss of generality, that each V_i is an independent set. We present a slight modification of a known reduction (see [17, Theorem 13.9]).
Reduction The reduction takes as input an instance $(G, q, (V_1, \dots, V_q))$ of Multicolored Independent Set and constructs a graph G' as follows.

- For every vertex $v \in V(G)$, the reduction introduces a vertex v into G': we denote by C the set of all these vertices in G'. Note that the sets V_i carry over directly to G'.
- The reduction turns the set C into a clique in G' by adding edges between any two distinct vertices of C.
- For every $i \in [q]$, the reduction introduces two new vertices x_i, y_i into G' and makes them adjacent to every vertex of V_i.
- For every edge $e = uv \in E(G)$ with endpoints $u \in V_i$ and $v \in V_j$, the reduction introduces a vertex w_e into G' and makes it adjacent to every vertex of $(V_i \cup V_j) \setminus \{u, v\}$.

For DomSet and connDomSet the reduction returns the instance (G', q). For Steiner Tree, it sets all the vertices in $V(G') \setminus C$ as terminals and returns the instance $(G', V(G') \setminus C, q)$.

Correctness In the following claim, we prove that the reduction produces equivalent instances. We only prove the claim for DomSet; the correctness for the other two problems follows immediately from the design of the graph G'.

Claim 6.3 $(G, q, (V_1, \dots, V_q))$ *is a* Yes-*instance for* Multicolored Independent Set *if and only if* (G', q) *is a* Yes-*instance for* DomSet.

Proof Assume that $(G, q, (V_1, \dots, V_q))$ is a Yes-instance for Multicolored Independent Set and let I be an independent set I of G containing one vertex from each V_i. We claim that I is a dominating set in G'. Since for every $i \in [q]$, $I \cap V_i \neq \emptyset$, the set I dominates every vertex in $V_i \cup \{x_i, y_i\}$. For an edge $e = uv \in E(G)$ where $u \in V_i$ and $v \in V_j$, consider the vertex w_e. As u and v are adjacent, at least one of them is not in I, say $u \notin I$ without loss of generality. Since $I \cap V_i \neq \emptyset$, there must then exist $w \in V_i \setminus \{u\}$ such that $w \in I$; but w_e is adjacent to w by construction and thus, I dominates w_e.

Conversely, assume that (G', q) is a Yes-instance for DomSet and let D be a dominating set of size q in G'. We claim that D is also an independent set in G. Since for every $i \in [q]$, D dominates the vertices x_i and y_i, D has to contain at least one vertex from $V_i \cup \{x_i, y_i\}$; and since x_i and y_i are not adjacent, in fact D must contain a vertex from V_i. As these sets are disjoint for different values of i and $|D| \leq q$, it follows that D contains exactly one vertex from each V_i: let $v_1 \in V_1, \dots, v_q \in V_q$ be the vertices of D. Now suppose for a contradiction that v_i and v_j are the endpoints of an edge e. By construction, vertex w_e in G' is adjacent only to $(V_i \cup V_j) \setminus \{v_i, v_j\}$ and hence, D does not dominate w_e, a contradiction.

The following claim holds for all three problems as it only depends on the structure of the graph G'.

Claim 6.4 *G' does not contain H_{2q+2} as an induced subgraph.*

Proof We first partition the vertex set of $V(G')$. For this, let $I = V(G') \setminus C$. It is easy to see that I is an independent set and since C is a clique in G', G' is in fact a split graph with split partition (C, I). For each integer $i \in [q]$, the vertices in I can be partitioned into the following three sets depending on their adjacency in V_i.

1. Vertices that are adjacent to *all* vertices in V_i: these are the vertices x_i, y_i.
2. Vertices that are adjacent to *all but one* vertex in V_i: these are the vertices of type w_e for edges e with one endpoint in V_i.
3. Vertices that are adjacent to *no* vertex in V_i: these are the vertices of type w_e for edges e with both endpoints are outside V_i, and the vertices $x_{i'}, y_{i'}$ where $i \neq i'$.

Recall that by assumption, V_i is an independent set in G and thus, there is no edge with both endpoints in V_i.

Now suppose, for the sake of contradiction, that G' contains H_{2q+2} as an induced subgraph. Consider the (unique) split partition (H_C, H_I) of H_{2q+2}. Let $H_C = \{v_1, v_2, \ldots, v_{2q+2}\}$ and $H_I = \{u_1, u_2, \ldots, u_{2q+2}\}$. Moreover, for every $i \in [2q+2]$, edge $v_i u_i$ is in $E(H_{2q+2})$. Consider the clique H_C in G'. As I is an independent set, $|H_C \cap I| \leq 1$. Hence, H_C contains at least $2q+1$ vertices of C. By the Pigeon-Hole principle, there must then exist an integer $i \in [q]$ such that $|H_C \cap V_i| \geq 3$: let v_1, v_2, v_3 be three vertices of $H_C \cap V_i$.

Since by construction, u_1 is not adjacent to v_2 and v_3, and v_2, v_3 are in C, it must be that $u_1 \in I$. But then, u_1 is adjacent to one vertex in V_i, namely v_1, and nonadjacent to two vertices in V_i, namely v_2 and v_3, a contradiction to the fact that vertices in I can be partitioned into the three sets described above. Therefore, G does not contain H_{2q+2} as an induced subgraph.

It is known that, assuming the ETH, there is no algorithm that can solve MULTICOLORED INDEPENDENT SET on instance $(G, q, (V_1, V_2, \ldots V_q))$ in time $f(q) \cdot |V(G)|^{o(q)}$ for any computable function f (see, e.g., [17, Corollary 14.23]). Note finally, that $|V(G')| \in \mathcal{O}(|V(G)|^2)$ and G' is an H_{2q+2} induced-subgraph-free split graph. These facts, together with arguments that are standard for parameter preserving reductions, concludes the proof of the lemma. □

In the following, we give the XP-algorithms for the three problems. Instead of giving the algorithm for DOMSET, we give an algorithm for the more general RED-BLUE- DOMSET. Recall that, from Lemma 3.1, there is a reduction from the former to the latter problem. There remains to argue that this reduction preserves the property of being H_ℓ-induced-subgraph-free.

Lemma 6.5 *There is a polynomial-time algorithm that given an instance (G, k) of* DOMSET *constructs an equivalent instance $(G', (R', B'), k)$ of* RED- BLUE- DOMSET *such that if G is a H_ℓ-induced-subgraph-free graph, then so is G'.*

Proof As in Lemma 3.1, we construct G' from G as follows. For every vertex $v \in V(G)$, add two copies v_R and v_B to $V(G')$ and add an edge $v_R v_B$ to $E(G')$. For every edge $uv \in E(G)$, add edges $v_R u_R$, $v_R u_B$, $v_B u_R$, and $v_B u_B$ to $E(G')$. This completes the construction of G'. By the proof of Lemma 3.1, it is known that these two instances are equivalent. In the following, we let $R' = \{v_R \mid v \in V(G)\}$ and $B' = \{v_B \mid v \in V(G)\}$.

Now assume that G is H_ℓ-induced-subgraph-free and suppose, for the sake of contradiction, that G' contains H_ℓ as an induced subgraph. Let I be the vertices forming the independent set and C the vertices forming the clique of H_ℓ. We claim that for no vertex $v \in V(G)$, we have that $v_B, v_R \in C \cup I$. Note that if the claim holds, then using the original version of each vertex would give an induced H_ℓ in G and thus contradict our assumption.

There remains to prove the claim. To this end, consider $v \in V(G)$. Since I is an independent set, v_B and v_R cannot both be contained in I. Moreover, it can also not be the case that $v_B \in I$ and $v_R \in C$ (or vice-versa) as then v_B would also be adjacent to all vertices in C. Hence, assume that $v_B, v_R \in C$. Assume, without loss of generality, that $u_B \in I$ is the unique vertex adjacent to v_B in C (the case where $u_R \in I$ is the unique adjacent vertex is symmetric). Since there is an edge from v_B to u_B, we know that u and v are adjacent in G. Hence, by construction, there must also be an edge from v_R to u_B which contradicts the fact that we have an H_ℓ graph. □

We are now ready to show that RED- BLUE- DOMSET on chordal graphs admits an XP-algorithm if the input graph does not contain H_ℓ as induced subgraph.

Lemma 6.6 RED- BLUE DOMINATING SET *restricted to H_ℓ-induced-subgraph-free chordal graphs admits an algorithm running in time $n^{\mathcal{O}(\ell)}$.*

Proof Let $(G, (R, B), k)$ be an instance of RED- BLUE- DOMSET where G is an H_ℓ-induced-subgraph-free chordal graph, and let $(T, \mathcal{M})$ be a tree representation of G. First, we add a node r to T by connecting it to an arbitrary node of T and root T at r (note that, by construction, no model in $\mathcal{M}$ contains r). We use dynamic programming to compute the entries of two tables T_1 and T_2 in a bottom-up traversal of T. The contents of T_1 and T_2 are defined as follows. For every node $\alpha \in V(T)$ and every *nonempty* set $X \subseteq R_\alpha^{\in}$ of size at most ℓ,

$$\mathsf{T}_1[\alpha, X] := \min\{|S| \mid S \subseteq R_\alpha^{\cap}, S \cap R_\alpha^{\in} = X, N[S] \supseteq B_\alpha^{\cap}\}$$

Intuitively, this stores the (size of the) smallest set of red vertices containing X such that all blue vertices in T_α are dominated.

For every node $\alpha \in V(T)$ and every set $Y \subseteq R_\alpha^{\subseteq\dagger}$ of size at most ℓ,

$$\mathsf{T}_2[\alpha, Y] := \min\{|S| \mid S \subseteq R_\alpha^{\subseteq\dagger}, N[S] \supseteq B_\alpha^{\subseteq\dagger} \cup (N(Y) \cap B_\alpha^{\in})\}$$

Intuitively, this stores the (size of the) smallest set of red vertices intersecting with T_α but not α which dominate all blue vertices below α and the α-blues that are neighbors of the red vertices in Y.

Initially, every entry of T_1 and T_2 is set to $+\infty$. The output is YES if and only if $\mathsf{T}_2[\mathsf{r}, \emptyset] \leq k$. We next show how to update the entries of T_1 and T_2.

Updating the Leaves Let $\alpha \in V(T)$ be a leaf of T. Then set

$$\mathsf{T}_2[\alpha, \emptyset] = 0$$

and for every nonempty set $X \subseteq R^{\in}_{\alpha}$ of size at most ℓ, set

$$\mathsf{T}_1[\alpha, X] = |X|.$$

Updating Internal Nodes Let $\alpha \in V(T)$ be an internal node of T and let $\beta_1, \dots, \beta_p$ be the children of α. To update the entries of $\mathsf{T}_1[\alpha, \cdot]$, we proceed as follows. Let $X \subseteq R^{\in}_{\alpha}$ be a nonempty set of size at most ℓ. Denote by $I \subseteq [p]$ the set of indices $i \in [p]$ such that $X \cap R^{\in}_{\beta_i} \neq \emptyset$ and set $\overline{I} = [p] \setminus I$. For every $i \in I$, further let $X_i = X \cap R^{\in}_{\beta_i}$. We update $\mathsf{T}_1[\alpha, X]$ according to the following procedure.

1. For every $i \in I$, set

$$m_i = \min_{\substack{Z \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha} \\ \text{s.t. } |Z|+|X_i| \leq \ell}} \mathsf{T}_1[\beta_i, Z \cup X_i].$$

2. For every $i \in \overline{I}$, let

$$\mathcal{Y}_i = \{Z \subseteq R^{\subseteq\dagger}_{\beta_i} \mid |Z| \leq \ell \text{ and } B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha} \subseteq N(Z)\}$$

and set

$$m^1_i = \min_{\substack{Z \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha} \\ \text{s.t. } 1 \leq |Z| \leq \ell}} \mathsf{T}_1[\beta_i, Z] \qquad \text{and} \qquad m^2_i = \min_{Z \in \mathcal{Y}_i} \mathsf{T}_2[\beta_i, Z].$$

3. Set

$$\mathsf{T}_1[\alpha, X] = |X| + \sum_{i \in I} m_i - |X_i| + \sum_{i \in \overline{I}} \min\{m^1_i, m^2_i\}.$$

To update the entries of $\mathsf{T}_2[\alpha, \cdot]$, we proceed as follows. Let $Y \subseteq R^{\subseteq\dagger}_{\alpha}$ be a set of size at most ℓ. Denote by $I \subseteq [p]$ the set of indices $i \in [p]$ such that $Y \cap R^{\cap}_{\beta_i} \neq \emptyset$ and set $\overline{I} = [p] \setminus I$. We update $\mathsf{T}_2[\alpha, Y]$ according to the following procedure.

1. Initialise $\mathsf{OPT}_{\overline{I}} = 0$ and $\mathsf{OPT}_I = +\infty$.
2. For every $i \in \overline{I}$ do:

   2.a. Let

$$\mathcal{Y}_i = \{Z \subseteq R^{\subseteq\dagger}_{\beta_i} \mid |Z| \leq \ell \text{ and } B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha} \subseteq N(Z)\}$$

```
    and set

    $$m_i^1 = \min_{\substack{Z \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha} \\ \text{s.t. } 1 \leq |Z| \leq \ell}} \mathsf{T}_1[\beta_i, Z] \qquad \text{and} \qquad m_i^2 = \min_{Z \in \mathcal{Y}_i} \mathsf{T}_2[\beta_i, Z].$$

  2.b. Set $\mathsf{OPT}_{\overline{I}} = \mathsf{OPT}_{\overline{I}} + \min\{m_i^1, m_i^2\}$.
3. For every partition $N = \{N_i \mid i \in I\}$ of $N(Y) \cap B^{\in}_{\alpha}$ where for every $i \in I$, $N_i \subseteq N(Y_i) \cap B^{\in}_{\alpha}$ do:
  3.a. Initialise $\mathsf{Int}_N = 0$.
  3.b. For every $i \in I$ do:
    3.b.i. Let

      $$\mathcal{Y}_i^N = \{Z \subseteq R^{\subseteq\dagger}_{\beta_i} \mid |Z| \leq \ell \text{ and } N_i \cup (B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha}) \subseteq N(Z)\}$$

      and set

      $$m_i^1 = \min_{\substack{Z \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha} \\ \text{s.t. } 1 \leq |Z| \leq \ell}} \mathsf{T}_1[\beta_i, Z] \text{ and } m_i^2 = \min_{Z \in \mathcal{Y}_i^N} \mathsf{T}_2[\beta_i, Z].$$

    3.b.ii. Set $\mathsf{Int}_N = \mathsf{Int}_N + \min\{m_i^1, m_i^2\}$.
  3.c. Set $\mathsf{OPT}_I = \min\{\mathsf{OPT}_I, \mathsf{Int}_N\}$.
4. Set $\mathsf{T}_2[\alpha, Y] = \mathsf{OPT}_{\overline{I}} + \mathsf{OPT}_I$.
```

We next show that the entries of $\mathsf{T}_1[\alpha, \cdot]$ and $\mathsf{T}_2[\alpha, \cdot]$ are updated correctly. To this end, we first introduce some useful notation. Given a set $X \subseteq B$, a set $S \subseteq R$ *minimally dominates* X if $X \subseteq N(S)$ and for every $x \in S$, $X \nsubseteq N(S \setminus \{x\})$. Additionally, we prove the following.

Claim 6.7 *For every node $\alpha \in V(T)$, the following hold.*

(i) For every minimum red-blue dominating set S of G, $|S \cap R^{\in}_{\alpha}| \leq \ell$.
(ii) For every set $X \subseteq B^{\in}_{\alpha}$ and every set $Y \subseteq R \setminus R^{\in}_{\alpha}$ minimally dominating X, $|Y| \leq \ell$.

Proof To prove item (i), let S be a minimum red-blue dominating set of G. Since S is minimum, for every $x \in S \cap R^{\in}_{\alpha}$, there exists $p_x \in N(x) \cap B$ such that $p_x \notin \bigcup_{y \in S \setminus \{x\}} N(y)$, i.e., the blue vertex p_x is only dominated by x. Then $\{p_x \mid x \in S \cap R^{\in}_{\alpha}\}$ is an independent set: indeed, if there exist $x, y \in S \cap R^{\in}_{\alpha}$ such that $p_x p_y \in E(G)$ then x, p_x, p_y, y induces a C_4, a contradiction as G is chordal. It follows that $(S \cap R^{\in}_{\alpha}) \cup \{p_x \mid x \in S \cap R^{\in}_{\alpha}\}$ induces an $H_{|S \cap R^{\in}_{\alpha}|}$ and so, $|S \cap R^{\in}_{\alpha}| < \ell$.

To prove item (ii), let $X \subseteq B^{\in}_{\alpha}$ and let $Y \subseteq R \setminus R^{\in}_{\alpha}$ be a set minimally dominating X. Since Y is minimal, for every $x \in Y$, there exists $p_x \in N(x) \cap X$ such that $p_x \notin \bigcup_{y \in Y \setminus \{x\}} N(y)$, i.e., the blue vertex p_x is only dominated by x. This implies that Y is an independent set: indeed, if there exist $x, y \in Y$ such that $xy \in E(G)$ then x, p_x, p_y, y induces a C_4, a contradiction as G is chordal. It follows that $Y \cup \{p_x \mid x \in Y\}$ induces an $H_{|Y|}$ and so, $|Y| < \ell$.

We now move towards proving the correctness of the update procedure. We start with the first table.

Claim 6.8 *For every internal node $\alpha \in V(T)$, the entries of $\mathsf{T}_1[\alpha, \cdot]$ are updated correctly. Furthermore, $\mathsf{T}_1[\alpha, \cdot]$ can be updated in $n^{\mathcal{O}(\ell)}$-time.*

Proof Let $\alpha \in V(T)$ be an internal node of T with children $\beta_1, \ldots, \beta_p$ and assume that for every $i \in [p]$, $\mathsf{T}_1[\beta_i, \cdot]$ and $\mathsf{T}_2[\beta_i, \cdot]$ have been correctly filled. Let us first show that for every nonempty set $X \subseteq R^{\in}_{\alpha}$ of size at most ℓ, there exists a set $S \subseteq R^{\cap}_{\alpha}$ of size $\mathsf{T}_1[\alpha, X]$ such that $S \cap R^{\in}_{\alpha} = X$ and S dominates every vertex in $B^{\in}_{\alpha}$.

Consider a nonempty set $X \subseteq R^{\in}_{\alpha}$ of size at most ℓ. Let $I \subseteq [p]$ be the set of indices $i \in [p]$ such that $X \cap R^{\in}_{\beta_i} \neq \emptyset$ and set $\overline{I} = [p] \setminus I$. For every $i \in I$, further let $X_i = X \cap R^{\in}_{\beta_i}$. For every $i \in I$, let m_i be as defined in Step 6.1 and let $Z_i \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha}$ be a set such that $|Z_i| + |X_i| \leq \ell$ and $m_i = \mathsf{T}_1[\beta_i, Z_i \cup X_i]$.

Then, since for every $i \in I$, $\mathsf{T}_1[\beta_i, \cdot]$ has been correctly filled, there exists a set $S_i \subseteq R^{\cap}_{\beta_i}$ of size $\mathsf{T}_1[\beta_i, Z_i \cup X_i]$ such that $S_i \cap R^{\in}_{\beta_i} = Z_i \cup X_i$ and S_i dominates every vertex in $B^{\cap}_{\beta_i}$. Similarly, for every $i \in \overline{I}$, let m^1_i and m^2_i be as defined in Step 6.1. Further let $\overline{I}_1 \subseteq \overline{I}$ be the set of indices $i \in \overline{I}$ such that $\min\{m^1_i, m^2_i\} = m^1_i$ and set $\overline{I}_2 = \overline{I} \setminus \overline{I}_1$. For every $i \in \overline{I}_1$, let $Z_i \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha}$ be a set such that $1 \leq |Z_i| \leq \ell$ and $m^1_i = \mathsf{T}_1[\beta_i, Z_i]$; and for every $i \in \overline{I}_2$, let $Z_i \subseteq R^{\subseteq\dagger}_{\beta_i}$ be a set of size at most ℓ such that $B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha} \subseteq N(Z_i)$ and $m^2_i = \mathsf{T}_2[\beta_i, Z_i]$.

Then, since for every $i \in \overline{I}_1$, $\mathsf{T}_1[\beta_i, \cdot]$ has been correctly filled, there exists a set $S_i \subseteq R^{\cap}_{\beta_i}$ of size $\mathsf{T}_1[\beta_i, Z_i]$ such that $S_i \cap R^{\in}_{\beta_i} = Z_i \cup X_i$ and S_i dominates every vertex in $B^{\cap}_{\beta_i}$; similarly, since for every $i \in \overline{I}_2$, $\mathsf{T}_2[\beta_i, \cdot]$ has been correctly filled, there exists a set $S_i \subseteq R^{\cap}_{\beta_i} \setminus R^{\in}_{\beta_i}$ of size $\mathsf{T}_2[\beta_i, Z_i]$ such that S_i dominates every vertex in $B^{\subseteq\dagger}_{\beta_i} \cup (N(Z_i) \cap B^{\in}_{\beta_i})$.

We contend that the set $M = X \cup \bigcup_{i \in [p]} S_i$ is the desired S. Indeed, observe first that, by the update step, $\mathsf{T}_1[\alpha, X] = |X| + \sum_{i \in I} |S_i| - |X_i| + \sum_{i \in \overline{I}} |S_i| = |M|$. Let us next show that $M \cap R^{\in}_{\alpha} = X$. Since for every $i \in I$, $\mathsf{T}_1[\beta_i, \cdot]$ is correctly filled, $S_i \cap R^{\in}_{\beta_i} = X_i \cup Z_i$ where $Z_i \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha}$ by construction; similarly, for every $i \in \overline{I}_1$, $S_i \cap R^{\in}_{\beta_i} = Z_i$ where $Z_i \cap R^{\in}_{\alpha} = \emptyset$ since $Z_i \subseteq R^{\subseteq\dagger}_{\beta_i}$ by definition. Now by construction, for every $i \in \overline{I}_2$, $S_i \cap R^{\in}_{\alpha} = \emptyset$ since $S_i \subseteq R^{\subseteq\dagger}_{\beta_i}$; thus, $M \cap R^{\in}_{\alpha} = \bigcup_{i \in I} X_i = X$ as claimed.

Let us finally show that M dominates every vertex in $B^{\cap}_{\alpha}$. First observe that since $X \neq \emptyset$, every vertex in $B^{\in}_{\alpha}$ is dominated by M. Consider therefore a vertex $x \in B^{\subseteq\dagger}_{\alpha}$. Then there exists $i \in [p]$ such that $x \in B^{\cap}_{\beta_i}$. If $i \in I$ then x is dominated by S_i by definition; similarly, if $i \in \overline{I}_1$ then x is dominated by S_i by definition. Thus, suppose that $i \in \overline{I}_2$. Then either $x \in B^{\subseteq\dagger}_{\beta_i}$ in which case x is dominated by S_i by definition; or $x \in B^{\in}_{\beta_i}$ and since $x \notin B^{\in}_{\alpha}$ by assumption, $x \in N(Z_i)$ by construction and thus, x is dominated by S_i by definition. Therefore, M dominates every vertex in $B^{\cap}_{\alpha}$ and so, M is indeed the desired S.

Consider now a minimum red-blue dominating set S of G such that $S \cap R^{\in}_{\alpha} \neq \emptyset$. Then by Claim 6.7(i), $|S \cap R^{\in}_{\alpha}| \leq \ell$. Let us show that $|S \cap R^{\cap}_{\alpha}| \geq \mathsf{T}_1[\alpha, S \cap R^{\in}_{\alpha}]$.

Denote by $X = S \cap R^{\in}_{\alpha}$ and for every $i \in [p]$, let $S_i = S \cap R^{\cap}_{\beta_i}$. Further let $I \subseteq [p]$ be the set of indices $i \in [p]$ such that $S_i \cap R^{\in}_{\beta_i} \neq \emptyset$ and set $\overline{I} = [p] \backslash I$. By Claim 6.7(i), for every $i \in I$, $|S_i \cap R^{\in}_{\beta_i}| \leq \ell$ and since $\mathsf{T}_1[\beta_i, \cdot]$ has been correctly filled, $|S_i| \geq \mathsf{T}_1[\beta_i, S_i \cap R^{\in}_{\beta_i}]$. Now consider $i \in \overline{I}$. Since S is dominating and $S_i \cap R^{\in}_{\beta_i} = \emptyset$, every vertex in $B^{\in}_{\beta_i} \backslash B^{\in}_{\alpha}$ must be dominated by some vertex in $S \cap R^{\subseteq\dagger}_{\beta_i}$: let $S^*_i \subseteq S \cap R^{\subseteq\dagger}_{\beta_i}$ be a set minimally dominating $B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha}$. Then by Claim 6.7(ii), $|S^*_i| \leq \ell$ and since $\mathsf{T}_2[\beta_i, \cdot]$ has been correctly filled, $|S_i| \geq \mathsf{T}_2[\beta_i, S^*_i]$. Thus, we conclude by the update step and the above that

$$\begin{aligned} \mathsf{T}_1[\alpha, X] &\leq |X| + \sum_{i \in I} \mathsf{T}_1[\beta_i, S_i \cap R^{\in}_{\beta_i}] - |S_i \cap X| + \sum_{i \in \overline{I}} \mathsf{T}_2[\beta_i, S^*_i] \\ &\leq |X| + \sum_{i \in I} |S_i| - |S_i \cap X| + \sum_{i \in \overline{I}} |S_i| = |S \cap R^{\cap}_{\alpha}| \end{aligned}$$

as claimed. Now by observing that $S \cap R^{\cap}_{\alpha}$ is a minimum-sized set dominating every vertex in $B^{\cap}_{\alpha}$ and whose intersection with $R^{\in}_{\alpha}$ is X (S would otherwise not be minimum), we conclude by the above that $\mathsf{T}_1[\alpha, \cdot]$ is updated correctly.

Finally, it is not difficult to see that it takes $n^{\mathcal{O}(\ell)}$-time to update one entry of $\mathsf{T}_1[\alpha, \cdot]$ and since there are $n^{\mathcal{O}(\ell)}$ entries, the claim follows.

We next show the correctness of the update procedure for the second table.

Claim 6.9 *For every internal node $\alpha \in V(T)$, the entries of $\mathsf{T}_2[\alpha, \cdot]$ are updated correctly. Furthermore, $\mathsf{T}_2[\alpha, \cdot]$ can be updated in $n^{\mathcal{O}(\ell)}$-time.*

Proof Let $\alpha \in V(T)$ be an internal node of T with children $\beta_1, \ldots, \beta_p$ and assume that for every $i \in [p]$, $\mathsf{T}_1[\beta_i, \cdot]$ and $\mathsf{T}_2[\beta_i, \cdot]$ have been correctly filled. Let us first show that for every set $Y \subseteq R^{\subseteq\dagger}_{\alpha}$ of size at most ℓ, there exists a set S of size $\mathsf{T}_2[\alpha, Y]$ such that S dominated every vertex in $B^{\subseteq\dagger}_{\alpha} \cup (N(Y) \cap B^{\in}_{\alpha})$.

Consider a set $Y \subseteq R^{\subseteq\dagger}_{\alpha}$ of size at most ℓ. Let $I \subseteq [p]$ be the set of indices $i \in [p]$ such that $Y \cap R^{\cap}_{\beta_i} \neq \emptyset$ and set $\overline{I} = [p] \setminus I$. For every $i \in \overline{I}$, let $\mathcal{Y}_i$, m^1_i and m^2_i be as defined in Step 6.1. Further let $\overline{I}_1 \subseteq \overline{I}$ be the set of indices $i \in \overline{I}$ such that $\min\{m^1_i, m^2_i\} = m^1_i$ and set $\overline{I}_2 = \overline{I} \setminus \overline{I}_1$. Let $N = \{N_i \mid i \in I\}$ be a partition of $N(Y) \cap B^{\in}_{\alpha}$ as considered in Step 6.1 such that the final value of Int_N is minimum among all such final values taken over every partition of $N(Y) \cap B^{\in}_{\alpha}$ as considered in Step 6.1. For every $i \in I$, let $\mathcal{Y}^N_i$, m^1_i and m^2_i be as defined in Step 6.1. Let $I_1 \subseteq I$ be the set of indices $i \in I$ such that $\min\{m^1_i, m^2_i\} = m^1_i$ and set $I_2 = I \setminus I_1$.

For every $i \in I_1 \cup \overline{I}_1$, let $Z_i \subseteq R^{\in}_{\beta_i} \setminus R^{\in}_{\alpha}$ be a nonempty set of size at most ℓ such that $m^1_i = \mathsf{T}_1[\beta_i, Z_i]$. Then, since for every $i \in I_1 \cup \overline{I}_1$, $\mathsf{T}_1[\beta_i, \cdot]$ has been correctly updated, there exists a set $S_i \subseteq R^{\cap}_{\beta_i}$ of size $\mathsf{T}_1[\beta_i, Z_i]$ such that $S_i \cap R^{\in}_{\beta_i} = Z_i$ and S_i dominates every vertex in $B^{\cap}_{\beta_i}$.

Now for every $i \in I_2$, let $Z_i \in \mathcal{Y}_i$ be a set of size at most ℓ such that $m^2_i = \mathsf{T}_2[\beta_i, Z_i]$; similarly, for every $i \in \mathcal{Y}^N_i$, let $Z_i \in \mathcal{Y}^N_i$ be a set of size at most ℓ such that $m^2_i = \mathsf{T}_2[\beta_i, Z_i]$. Then, since for every $i \in I_2 \cup \overline{I}_2$, $\mathsf{T}_2[\beta_i, \cdot]$ has been correctly filled, there exists a set $S_i \subseteq R^{\subseteq\dagger}_{\beta_i}$ of size $\mathsf{T}_2[\beta_i, Z_i]$ such that S_i dominates every vertex in $B^{\subseteq\dagger}_{\beta_i} \cup (N(Z_i) \cap B^{\in}_{\beta_i})$.

We contend that the set $M = \bigcup_{i\in I} S_i$ is the desired S. Indeed, observe first that, by the update step, $\mathsf{T}_2[\alpha, Y] = \sum_{i\in\overline{I}} |S_i| + \sum_{i\in I} |S_i| = |M|$. Now consider a vertex $x \in B^{\subseteq\dagger}_{\alpha} \cup (N(Y) \cap B^{\in}_{\alpha})$ and let us show that x is dominated by M.

Suppose first that $x \notin B^{\in}_{\alpha}$. Then there exists $i \in I$ such that $x \in B^{\cap}_{\beta_i}$. If $i \in I_1 \cup \overline{I}_1$ then x is dominated by S_i by definition. Suppose therefore that $i \in I_2 \cup \overline{I}_2$. If $x \in B^{\subseteq\dagger}_{\beta_i}$ then x is dominated by S_i by definition; otherwise, $x \in B^{\in}_{\beta_i}$ and since $x \notin B^{\in}_{\alpha}$ by assumption, $x \in N(Z_i)$ by construction and so, x is dominated by S_i by definition.

Suppose second that $x \in N(Y) \cap B^{\in}_{\alpha}$. Then there exists $i \in I$ such that $x \in N_i$. If $i \in I_1$ then x is dominated by S_i by definition. Suppose therefore that $i \in I_2$. If $x \in B^{\subseteq\dagger}_{\beta_i}$ then x is dominated by S_i by definition; otherwise, $x \in B^{\in}_{\beta_i}$ and since $x \in N_i \subseteq N(Z_i)$ by construction, x is dominated by S_i by definition. Therefore, M dominates every vertex $B^{\subseteq\dagger}_{\alpha} \cup (N(Y) \cap B^{\in}_{\alpha})$ and so, M is indeed the desired S.

Consider now a minimum red-blue dominating set S of G such that $S \cap R^{\in}_{\alpha} = \emptyset$ and let $Y \subseteq S \cap R^{\subseteq\dagger}_{\alpha}$ be a set minimally dominating $N(S \cap R^{\subseteq\dagger}_{\alpha}) \cap B^{\in}_{\alpha}$. Then by Claim 6.7(ii), $|Y| \leq \ell$. Let us show that $|S \cap R^{\subseteq\dagger}_{\alpha}| \geq \mathsf{T}_2[\alpha, Y]$. For every $i \in [p]$, let $S_i = S \cap R^{\cap}_{\beta_i}$. Further let $I \subseteq [p]$ be the set of indices $i \in [p]$ such that $Y \cap R^{\cap}_{\beta_i} \neq \emptyset$ and set $\overline{I} = [p] \setminus I$. By construction, for every $x \in N(Y) \cap B^{\in}_{\alpha}$, there exists $i \in I$ such that $x \in N(Y \cap R^{\cap}_{\beta_i})$: let $N = \{N_i \mid i \in I\}$ be a partition of $N(Y) \cap B^{\in}_{\alpha}$ where for every $i \in I$, $N_i \subseteq N(Y \cap R^{\cap}_{\beta_i})$.

Let $I_1 \subseteq I$ be the set of indices $i \in I$ such that $S_i \cap R^{\in}_{\beta_i} \neq \emptyset$ and set $I_2 = I \setminus I_1$. Then for every $i \in I_1$, $|S_i \cap R^{\in}_{\beta_i}| \leq \ell$ by Claim 6.7(i) and since $\mathsf{T}_1[\beta_i, \cdot]$ has been correctly filled, $|S_i| \geq \mathsf{T}_1[\beta_i, S_i \cap R^{\in}_{\beta_i}]$. Now for every $i \in I_2$, let $Z_i \subseteq R^{\subseteq\dagger}_{\beta_i}$ be a set minimally dominating $N_i \cup (B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha})$. Then for every $i \in I_2$, $|Z_i| \leq \ell$ by Claim 6.7(ii) (note indeed that $N_i \subseteq B^{\in}_{\beta_i}$) and since $\mathsf{T}_2[\beta_i, \cdot]$ has been correctly filled, $|S_i| \geq \mathsf{T}_2[\beta_i, Z_i]$.

Similarly, let $\overline{I}_1 \subseteq \overline{I}$ be the set of indices $i \in \overline{I}$ such that $S_i \cap R^{\in}_{\beta_i} \neq \emptyset$ and set $\overline{I}_2 = \overline{I} \setminus \overline{I}_2$. Then for every $i \in \overline{I}_1$, $|S_i \cap R^{\in}_{\beta_i}| \leq \ell$ by Claim 6.7(i) and since $\mathsf{T}_1[\beta_i, \cdot]$ has been correctly filled, $|S_i| \geq \mathsf{T}_1[\beta_i, S_i \cap R^{\in}_{\beta_i}]$. Now for every $i \in \overline{I}_2$, let $Z_i \subseteq R^{\subseteq\dagger}_{\beta_i}$ be a set minimally dominating $B^{\in}_{\beta_i} \setminus B^{\in}_{\alpha}$. Then for every $i \in \overline{I}_2$, $|Z_i| \leq \ell$ by Claim 6.7(ii) and since $\mathsf{T}_2[\beta_i, \cdot]$ has been correctly filled, $|S_i| \geq \mathsf{T}_2[\beta_i, Z_i]$. Thus, we conclude by the update step and the above that

$$\begin{aligned}\mathsf{T}_2[\alpha, Y] &\leq \sum_{i\in I_1\cup\overline{I}_1} \mathsf{T}_1[\beta_i, S_i \cap R^{\in}_{\beta_i}] + \sum_{i\in I_2\cup\overline{I}_2} \mathsf{T}_2[\beta_i, Z_i] \\ &\leq \sum_{i\in I} |S_i| = |S \cap R^{\subseteq\dagger}_{\alpha}|\end{aligned}$$

as claimed. Now by observing that $S \cap R^{\subseteq\dagger}_{\alpha}$ is a minimum-sized set dominating every vertex in $B^{\subseteq\dagger}_{\alpha} \cup (N(S \cap R^{\subseteq\dagger}_{\alpha}) \cap B^{\in}_{\alpha})$ (S would otherwise not be minimum), we conclude by the above that $\mathsf{T}_2[\alpha, \cdot]$ is updated correctly.

Finally, it is not difficult to see that Step 6.1 can be done in $n^{\mathcal{O}(\ell)}$-time and that, similarly, for a fixed partition, Steps 6.1–6.1 can be done in $n^{\mathcal{O}(\ell)}$-time. Now observe that $|I| \leq \ell$ since $|Y| \leq \ell$ and thus, there are at most $n^{\mathcal{O}(\ell)}$ partitions to consider in Step 6.1.

The lemma now follows from Claims 6.8 and 6.9. □

6.2 MultiCut with Undeletable Terminals

We present a simple reduction from Vertex Cover to Multicut with UnDel Term to prove Theorem 1.5. Consider an instance (G, q) of Vertex Cover where G has n vertices. Let G' be a graph obtained from a star with center r and $n+1$ leaves by subdividing each of its edge once. Fix an injective mapping $f : V(G) \mapsto V(G')$ such that $f(v)$ is a leaf for every $v \in V(G)$. Let w be the unique leaf which is not in the range of f. Then, the set of terminal pairs $\mathcal{P}$ is defined as follows: $\mathcal{P} = \{(f(u), f(v) \mid uv \in E(G)\} \cup \{(r, w)\}$. It is easy to see that (G, q) is a yes-instance of Vertex Cover if and only if $(G', \mathcal{P}, q)$ has a multicut of size at most q. As G' is acyclic, it is H_3-induced free.

7 Other Domination-Related Problems

The aim of this section is to complete the proofs of Theorem 1.1 and Theorem 1.4. More precisely, we show that Connected Red- Blue- DomSet and Steiner Tree are FPT parameterized by leafage and admit a $n^{\mathcal{O}(\ell)}$-algorithm on H_ℓ-induced-subgraph-free chordal graphs. The two problems are considered in two separate subsections.

7.1 Connected Red-Blue Dominating Set

In this subsection, we aim to prove that Connected Dominating Set is FPT parameterized by the leafage and admits a $n^{\mathcal{O}(\ell)}$-algorithm on H_ℓ-induced-subgraph-free chordal graphs. Formally, we prove the following.

Lemma 7.1 Connected Red- Blue Dominating Set *is* FPT *parameterized by the leafage and admits $n^{\mathcal{O}(\ell)}$-algorithm on H_ℓ-induced-subgraph-free chordal graphs.*

To obtain these results, we reduce in both cases to Red- Blue- DomSet and use the algorithms from Sect. 3 and Theorem 1.4, respectively. We describe below the reduction and show thereafter that both parameters are preserved. We first start with some useful terminology.

A tree representation $(T, \mathcal{M})$ of a graph G is *minimal* if for every edge $\alpha\beta \in E(T)$, the sets $\{x \in V(G) \mid \alpha \in \mathcal{M}(x)\}$ and $\{x \in V(G) \mid \beta \in \mathcal{M}(x)\}$ are inclusion-wise incomparable. A tree representation can easily be made minimal by contracting each edge $\alpha\beta \in E(T)$ for which the sets $\{x \in V(G) \mid \alpha \in \mathcal{M}(x)\}$ and $\{x \in V(G) \mid \beta \in \mathcal{M}(x)\}$ are inclusion-wise comparable. Note that this operation does not increase the number of leaves of the tree representation.

Reduction Let $(G, (R_G, B_G), k)$ be an instance of CONNECTED RED- BLUE- DOMSET and let $(T, \mathcal{M})$ be a minimal tree representation of G with $\mathtt{lf}(G)$ leaves. We construct an instance $(H, (R_H, B_H), k)$ of RED- BLUE- DOMSET as follows. More precisely, we construct a tree representation $(T_H, \mathcal{M}_H)$ for H by modifying $(T, \mathcal{M})$.

A node $\alpha \in V(T)$ is called a *red node* if $\alpha \notin \bigcup_{x \in B} \mathcal{M}(x)$, that is, α is contained only in models of red vertices.

Let T_B be the forest obtained by removing every red node in T and let $\overline{T}_B$ be the smallest connected subtree of T containing T_B. We further reduce $\overline{T}_B$ according to the following procedure.

- For each leaf α of $\overline{T}_B$ do:
  - Let $\beta \in V(\overline{T}_B)$ be the neighbor of α.
  - If $\{x \in B_G \mid \alpha \in \mathcal{M}(x)\} \subseteq \{x \in B_G \mid \beta \in \mathcal{M}(x)\}$, then set $\overline{T}_B = \overline{T}_B/\alpha\beta$.

Once the above procedure has been applied to $\overline{T}_B$, we subdivide each edge of $\overline{T}_B$ and let T_H be the resulting tree. We now define the vertices and edges of H as follows.

- For each node $\alpha \in V(T_H)$, we add a blue vertex x to B_H with model $\mathcal{M}_H(x) = \{\alpha\}$.
- For each red vertex $x \in R_G$ such that $\mathcal{M}(x) \cap V(\overline{T}_B) \neq \emptyset$, we add a red vertex r_x to R_H whose model $\mathcal{M}_H(r_x)$ in T_H corresponds to the subdivision of $\mathcal{M}(x) \cap V(\overline{T}_B)$.

We next show that these two instances are equivalent.

Lemma 7.2 *If $(G, (R_G, R_B), k)$ is a* YES*-instance for* CONNECTED RED- BLUE- DOMSET *then $(H, (R_H, B_H), k)$ is a* YES*-instance for* RED- BLUE- DOMSET.

Proof Let $D_G \subseteq R_G$ be a connected red-blue dominating set of G of size at most k and let $D_H = \{r_x \in R_H \mid x \in D_G \text{ and } \mathcal{M}(x) \cap V(\overline{T}_B) \neq \emptyset\}$. We contend that D_H is a red-blue dominating set of H.

Indeed, consider a blue vertex $x \in B_H$. By construction, there exists a node $\alpha \in V(T_H)$ such that $\mathcal{M}_H(x) = \{\alpha\}$. If α corresponds to a node or an edge of $\overline{T}_B - V(T_B)$ then, since D_G is connected, there exists $y \in D_G$ such that $\mathcal{M}(y)$ contains the node or edge corresponding to α; but then, $r_y \in D_H$ by construction and so, x is dominated. We conclude similarly if α corresponds to an edge between $V(\overline{T}_B) \setminus V(T_B)$ and $V(T_B)$.

Assume therefore that α corresponds to a node or an edge of T_B. Suppose first that α is a leaf of $\overline{T}_B$ and let β be the neighbor of α in $\overline{T}_B$. Then by construction, $\{x \in B_G \mid \alpha \in \mathcal{M}(x)\} \setminus \{x \in B_G \mid \beta \in \mathcal{M}(x)\} \neq \emptyset$ and so, there exists $y \in D_G$ such that $\alpha \in \mathcal{M}(y)$ since D_G is dominating; but then, $r_y \in D_H$ by construction and so, x is dominated. Suppose finally that α corresponds to an edge or an internal node of T_B. Since D_G is dominating and connected, there then exists $y \in D_G$ such that $\alpha \in \mathcal{M}(y)$; but then, $r_y \in D_H$ by construction and so, x is dominated. Therefore, D_H is a red-blue dominating set of H and since $|D_H| \leq k$, we conclude that $(H, (R_H, B_H), k)$ is a YES-instance for RED- BLUE- DOMSET. □

Lemma 7.3 *If $(H, (R_H, B_H), k)$ is a* YES*-instance for* RED- BLUE- DOMSET, *then $(G, (R_G, R_B), k)$ is a* YES*-instance* CONNECTED RED- BLUE- DOMSET.

Proof Let $D_H \subseteq R_H$ be a red-blue dominating set of H of size at most k and let $D_G = \{x \in R_G \mid r_x \in D_H\}$. We contend that D_G is a connected red-blue dominating set of G. Indeed, consider a blue vertex $x \in B_G$. Then by construction, there exists a node $\alpha \in V(\overline{T}_B)$ such that $\alpha \in \mathcal{M}(x)$. Since D_H is dominating, there then exists a vertex $r_y \in D_H$ such that the node in T_H corresponding to α is contained in $\mathcal{M}_H(r_y)$; but then, x is dominated since $y \in D_G$ and $\alpha \in \mathcal{M}(y)$ by construction. Now to see that D_G is connected, observe that if it weren't the case, there would exist an edge $\alpha\beta \in E(\overline{T}_B)$ such that no model in $\{\mathcal{M}(y) \mid y \in D_G\}$ contains the edge $\alpha\beta$; but then, the vertex in T_H corresponding to the edge $\alpha\beta$ wouldn't be dominated by D_H, a contradiction. Therefore, D_G is a connected red-blue dominating set of G as claimed and $|D_G| \leq k$. □

Finally, let us show that both parameters are preserved in the above reduction. First, it is not difficult to see that the leafage of H is at most that of G since the number of leaves of T_H is at most the number of leaves of T. Assume second that G is H_ℓ-induced-subgraph-free for some $\ell \geq 3$, and suppose for a contradiction that H contains an induced H_ℓ. Let $v_1, u_1, \ldots, v_\ell, u_\ell \in V(H)$ be 2ℓ such that $H[\{v_i u_i \mid i \in [\ell]\}]$ is isomorphic to H_ℓ where $\{v_i \mid i \in [\ell]\}$ is a clique and for every $i \in [\ell]$, $u_i v_i \in E(H)$. Note that since B_H is an independent set of H and every vertex in B_H is simplicial in H, $\{v_i \mid i \in [\ell]\} \cap B_H = \emptyset$; in particular, $\{v_i \mid i \in [\ell]\} \subseteq R_H \subseteq R_G$. Now for every $i \in [\ell]$, let α_i be a node of T_H defined as follows:

- If there is a node in $\mathcal{M}_H(u_i) \cap \mathcal{M}_H(v_i)$ which correspond to a node in T, then let α_i be any such node.
- Otherwise, $\mathcal{M}_H(u_i) \cap \mathcal{M}_H(v_i)$ contains only one node (namely, a node corresponding to an edge of T), in which case we let $\alpha_i \in \mathcal{M}_H(v_i)$ be the neighbor in $\mathcal{M}_H(v_i)$ of the node in $\mathcal{M}_H(u_i) \cap \mathcal{M}_H(v_i)$.

Note that, by construction, for every $i \in [\ell]$, α_i corresponds to a node of T which is, furthermore, contained in $\mathcal{M}(v_i)$. We contend that for every $i \in [\ell]$, there exists $x_i \in V(G) \backslash \{v_j \mid j \in [\ell]\}$ such that $x_i v_i \in E(G)$ and x_i is nonadjacent to $\{v_j \mid j \in [\ell] \backslash \{i\}\}$ in G, that is, $\{v_i, x_i \mid i \in [\ell]\}$ induces an H_ℓ in G. If true, this would contradict the fact that G is H_ℓ-induced-subgraph-free and thus conclude the proof. Let $i \in [\ell]$ and consider a node $\alpha \in \bigcap_{j \in [\ell]} \mathcal{M}(v_j)$ (note that since subtrees in a tree satisfy the Helly property, this intersection is nonempty). Further let $\beta \in V(T)$ be the neighbor of α_i on the path in T from α_i to α. Then since T is minimal, $I = \{x \in V(G) \mid \alpha_i \in \mathcal{M}(x)\} \backslash \{x \in V(G) \mid \beta \in \mathcal{M}(x)\} \neq \emptyset$; and since $\alpha_i, \beta \in \mathcal{M}(v_i)$, $v_i \notin I$. Thus, we may set $x_i = x$ where $x \in I$.

7.2 Steiner Tree

The aim of this section is to prove that Steiner Tree is FPT parameterized by the leafage and admits an $n^{\mathcal{O}(\ell)}$-algorithm on H_ℓ-induced-subgraph-free chordal graphs. To obtain these results, we give two parameter preserving reductions to Red-Blue-DomSet. We first present a general reduction rule for Steiner Tree instances.

Reduction Rule 7.4 *Let* $(G, \mathcal{T}, k)$ *be an instance of* Steiner Tree. *If* $G[\mathcal{T}]$ *has a connected component* C *of size greater than 1, then return the instance* $(G/V(C), (\mathcal{T} \setminus$

$V(C)) \cup \{v_C\}, k - |V(C)| + 1)$ where v_C is the vertex resulting from the contraction of C in G.

Lemma 7.5 *Reduction Rule 7.4 is safe. Furthermore, the leafage of $G/V(C)$ is at most that of G.*

Proof Suppose that such a connected component C exists. Assume first that $(G, \mathcal{T}, k)$ is a Yes-instance for Steiner Tree and let S be a solution for $(G, \mathcal{T}, k)$ such that the number of connected component in $S[V(C)]$ is minimum amongst all solutions for $(G, \mathcal{T}, k)$. We claim that $S[V(C)]$ has only one connected component. Indeed, suppose to the contrary that $S[V(C)]$ has at least two connected components. Since C is connected, there exist two connected components C_1 and C_2 of $S[V(C)]$ such that C_1 and C_2 are adjacent, that is, there is an edge $xy \in E(G)$ where $x \in V(C_1)$ and $y \in V(C_2)$. Let $L = z_1 \dots z_p$ be a shortest path in S from C_1 to C_2. Then the tree $S' = S - \{z_1z_2\} + \{xy\}$ is a solution for $(G, \mathcal{T}, k)$ such that $S'[V(C)]$ contains fewer connected component than $S[V(C)]$, a contradiction to the choice of S. Thus, $S[V(C)]$ has only one connected component and it is easy to see that $S/V(C)$ is a solution for $(G/V(C), (\mathcal{T} \setminus V(C)) \cup \{v_C\}, k - |V(C)| + 1)$.

Conversely, assume that $(G/V(C), (\mathcal{T} \setminus V(C)) \cup \{v_C\}, k - |V(C)| + 1)$ is a Yes-instance for Steiner Tree and let S be a solution. By construction, for every neighbor y of v_C in S, there exists $x \in V(C)$ such that $y \in N(x)$: for every $y \in N(v_C) \cap S$, let $x_y \in V(C)$ be an arbitrary vertex such that $y \in N(x_y)$. Set $V = \{x_y \mid y \in N(v_C) \cap S\}$ and for every $x \in V$, denote by $N_x = \{y \in N(v_C) \cap S \mid x_y = x\}$. Now let $x_1, \dots, x_p$ be an arbitrary ordering of V and let $y_1, \dots, y_q$ be an arbitrary ordering of $V(C) \setminus V$. Then the tree obtained from S by removing the vertex v_C to replace it with the path $x_1 \dots x_p y_1 \dots y_q$ and adding the edges $\{x_i z \mid i \in [p] \text{ and } z \in N_{x_i}\}$ is readily seen to be a solution for $(G, \mathcal{T}, k)$.

Finally, let us remark that a tree representation for $G/V(C)$ can be obtained from a tree representation $(T, \mathcal{M})$ of G by merging the models in $\{\mathcal{M}(x) \mid x \in V(C)\}$ into a single model representing v_C; in particular, the leafage of $G/V(C)$ is at most that of G. □

Lemma 7.6 *Let $(G, \mathcal{T}, k)$ be an instance of* Steiner Tree *and let $(G_R, \mathcal{T}_R, k)$ be the instance resulting from an exhaustive application of Reduction Rule 7.4 to $(G, \mathcal{T}, k)$. If G is H_ℓ-induced-subgraph-free then G_R is $H_{\ell+1}$-induced-subgraph-free.*

Proof Assume that $G[\mathcal{T}]$ contains at least one connected component of size greater than 1 (the lemma is trivial otherwise) and let $C_1, \dots, C_p$ be all such connected components of $G[\mathcal{T}]$. For every $i \in [p]$, denote by $v_{C_i} \in V(G_R)$ the vertex resulting from the contraction of C_i. Now assume that G is H_ℓ-induced-subgraph-free and suppose for a contradiction that G_R contains an induced $H_{\ell+1}$. Let $v_1, u_1, \dots, v_{\ell+1}, u_{\ell+1} \in V(G_R)$ be $2(\ell + 1)$ vertices inducing an $H_{\ell+1}$ in G_R where $\{v_i \mid i \in [\ell + 1]\}$ is the clique and for every $i \in [\ell+1]$, $v_iu_i \in E(G_R)$. Since $\{v_{C_i} \mid i \in [p]\}$ is an independent set in G_R, $|\{v_i \mid i \in [\ell + 1]\} \cap \{v_{C_i} \mid i \in [p]\}| \leq 1$: let us assume without loss of generality that $\{v_i \mid i \in [\ell]\} \cap \{v_{C_i} \mid i \in [p]\} = \emptyset$. On the other hand, if $v_{C_i} = u_{j_i}$ for some $i \in [p]$ and $j_i \in [\ell]$, then, by construction, there exists $x_i \in V(C_i)$ such that $x_iv_{j_i} \in E(G)$: let $I \subseteq [p]$ be the set of such indices. Then $X = \{x_i \mid i \in I\} \cup \{u_i \mid i \in [\ell] \setminus I\}$ is an independent set in G where each vertex in X has exactly one neighbor in $K = \{v_i \mid i \in [\ell]\}$, that is, $K \cup X$ induces an H_ℓ in G, a contradiction. □

Lemma 7.7 Steiner Tree *parameterized by the leafage is* FPT.

Proof As mentioned above, we reduce to Connected Red- Blue- DomSet: given an instance $(G, \mathcal{T}, k)$ of Steiner Tree, we construct an instance $(H, (R, B), k_H)$ of Connected Red- Blue- DomSet as follows. First, we assume that Reduction Rule 7.4 has been exhaustively applied to $(G, \mathcal{T}, k)$. This implies, in particular, that $\mathcal{T}$ is an independent set of G. Let us further assume that $|\mathcal{T}| > 1$ (the problem is trivial otherwise). Now let G^* be the supergraph of G obtained by making each terminal simplicial, that is, for every $t \in \mathcal{T}$, the neighborhood $N_G(t)$ $(= N_{G^*}(t))$ of t induces a clique in G^*. Observe that the leafage of G^* is at most that of G: indeed, a tree representation for G^* can be obtained from a tree representation $(T, \mathcal{M})$ of G as follows. For every terminal $t \in \mathcal{T}$, let $\alpha_t \in V(T)$ be a node of T contained the model $\mathcal{M}(t)$ of t. If there exists a neighbor $x \in N_{G^*}(t)$ such that $\mathcal{M}(x)$ does not contain α_t, then we extend $\mathcal{M}(x)$ by adding to it the path in T from α_t to α_x where $\alpha_x \in \mathcal{M}(x)$ is the closest node to α_t in T. By iterating this process and leaving all the other models intact, we obtain a tree representation $(T^*, \mathcal{M}^*)$ for G^* where T^* has the same number of leaves as T.

Reduction We may now construct the graph $(H, (R, B))$: the set $R = \{r_x \mid x \in V(G)\}$ of red vertices contains a copy of each vertex in $V(G)$ and the set $B = \{b_t \mid t \in \mathcal{T}\}$ of blue vertices contains a copy of each terminal. The graph $H[R]$ is then isomorphic to G^* and for every $t \in \mathcal{T}$, b_t is a true twin to r_t. Finally, we set $k_H = k - |\mathcal{T}|$. We next show that the instances $(G, \mathcal{T}, k)$ and $(H, (R, B), k_H)$ are equivalent.

Claim 7.8 *If* $(G, \mathcal{T}, k)$ *is a* Yes*-instance for* Steiner Tree*, then* $(H, (R, B), k_H)$ *is a* Yes*-instance for* Connected Red- Blue- DomSet.

Proof Assume that $(G, \mathcal{T}, k)$ is a Yes-instance for Steiner Tree and let S be a solution. Note that since $|\mathcal{T}| > 1$ by assumption, necessarily $V(S) \setminus \mathcal{T} \neq \emptyset$. We contend that the set $D = \{r_x \mid x \in V(S) \setminus \mathcal{T}\}$ is a solution for $(H, (R, B), k_H)$. Indeed, it is clear that for every $t \in \mathcal{T}$, b_t has a neighbor in D. To see that D is connected, observe that if a terminal $t \in \mathcal{T}$ is not a leaf of S, then t has at least two neighbors in S; but the neighborhood of r_t (and b_t) in R is clique in H and so, D is connected. Since $|D| \leq k - |\mathcal{T}| = k_H$, we conclude that D is indeed a solution for $(H, (R, B), k_H)$.

Claim 7.9 *If* $(H, (R, B), k_H)$ *is a* Yes*-instance for* Connected Red- Blue- DomSet*, then* $(G, \mathcal{T}, k)$ *is a* Yes*-instance for* Steiner Tree.

Proof Assume that $(H, (R, B), k_H)$ is a Yes-instance for Connected Red- Blue- DomSet and let D be a minimal solution. We contend that the set $S = \{x \mid r_x \in D\} \cup \mathcal{T}$ contains a solution for $(G, \mathcal{T}, k)$, that is, for every $t, t' \in \mathcal{T}$, there is a path from t to t' in $G[S]$. Observe first that for every $t \in \mathcal{T}$, $r_t \notin D$: indeed, if there exists $t \in \mathcal{T}$ such that $r_t \in D$, then surely r_t has at least one neighbor in D since $|B| = |\mathcal{T}| > 1$ and B is an independent set in H; but $N[b_t] = N[r_t]$ and $N[r_t]$ is a clique and so, $D \setminus \{r_t\}$ is still a solution for $(H, (R, B), k_H)$, a contradiction to the minimality of D. This implies, in particular, that $|S| = |D| + |\mathcal{T}| \leq k_H + |\mathcal{T}| = k$. Now since D is dominating and connected, for every terminal $t, t' \in \mathcal{T}$, there exists a path $P_{t,t'}$ in $H[D \cup \{b_t, b_{t'}\}]$ from b_t to $b_{t'}$; but then, it is easy to see that the set $\{x \in V(G) \mid r_x \in P_{t,t'}\} \cup \{t'' \in \mathcal{T} \mid V(P_{t,t'}) \cap N(r_{t''}) \neq \emptyset\} \subseteq S$ contains a path from t to t'. Therefore, S is a solution for $(G, \mathcal{T}, k)$.

Observe finally that a tree representation for $(H, (R, B))$ can be obtained from the tree representation $(T^*, \mathcal{M}^*)$ of G^* by adding a copy of $\mathcal{M}^*(t)$ for each terminal $t \in \mathcal{T}$; in particular, the leafage of H is at most that of G^* which concludes the proof. □

Lemma 7.10 *For every $\ell \geq 3$,* Steiner Tree *admits a $n^{\mathcal{O}(\ell)}$-algorithm on H_ℓ-induced-subgraph-free chordal graphs.*

Proof As mentioned above, we reduce to Connected Red-Blue-DomSet: given an instance $(G, \mathcal{T}, k)$ of Steiner Tree where G is an H_ℓ-induced-subgraph-free chordal graph, we construct an instance $(H, (R, B), k_H)$ of Connected Red-Blue-DomSet as follows. First, we assume that Reduction Rule 7.4 has been exhaustively applied to $(G, \mathcal{T}, k)$. This implies, in particular, that $\mathcal{T}$ is an independent set of G. Furthermore, by Lemma 7.6, G is $H_{\ell+1}$-induced-subgraph-free. Now the set $R = \{r_x \mid x \in V(G)\}$ of red vertices contains a copy of each vertex in $V(G)$ and the set $B = \{b_t \mid t \in \mathcal{T}\}$ of blue vertices contains a copy of each terminal. The graph $H[R]$ is then isomorphic to G and for every $t \in \mathcal{T}$, b_t is adjacent to only r_t. Furthermore, we set $k_H = k$. Now it is not difficult to see that these two instances are indeed equivalent: if S is a Steiner tree for $\mathcal{T}$ in G then $\{r_x \mid x \in V(S)\}$ is a connected red-blue dominating set of H; and conversely, if D is a connected red-blue dominating set then for every $t \in \mathcal{T}, r_t \in D$ and so, $\{x \mid r_x \in D\}$ contains a Steiner tree for $\mathcal{T}$. Finally, it is easily seen that H is $H_{\ell+2}$-induced-subgraph-free since $\{r_t \mid t \in \mathcal{T}\}$ is also an independent set in H and for every $t \in \mathcal{T}$, $N_H(b_t) = \{r_t\}$ (recall that G is $H_{\ell+1}$-induced-subgraph-free after the exhaustive application of Reduction Rule 7.4), which concludes the proof. □

8 Conclusion

In this article, we presented improved and new results regarding domination and cut problems on chordal graphs with bounded leafage. We presented an FPT algorithm running in time $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$-time for the Dominating Set problem on chordal graphs, and used it to obtain similar results for the Connected Dominating Set and Steiner Tree problems. Regarding cut problems, we proved that MultiCut with Undeletable Terminals on chordal graphs is W[1]-hard when parameterized by the leafage. We also presented a polynomial-time algorithm for Multiway Cut with Undeletable Terminals on chordal graphs. We find it surprising that the complexity of this problem was not known before. Finally, we examined these problems on H_ℓ-induced-subgraph-free chordal graphs to check the extent of our approach.

In the case of chordal graphs, we believe the leafage to be a more natural parameter than other popular parameters such as vertex cover, feedback vertex set or treewidth. It would be interesting to examine the structural parameterized complexity of problems such as Longest Cycle, Longest Path, Component Order Connectivity, s-Club Contraction, Independent Set Reconfiguration, Bandwidth, or Cluster Vertex Deletion. These problems are known to be NP-complete on split graphs and admit polynomial-time algorithms on interval graphs. Hence it is plausible that they admit an FPT or XP algorithm on chordal graphs parameterized by the leafage. We believe it is a representative list, though not exhaustive, of problems that exhibit this behavior. In fact, it would be fascinating to find a natural problem that does not exhibit this behavior, i.e., a problem that is NP-complete on interval graphs but admits a polynomial-time algorithm on split graphs.

Author Contributions All authors contributed to the study conception and design. All authors contributed to the writing of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. Research supported by the European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH. *Esther Galby and Prafullkumar Tale*: Part of the work was carried out when the authors were Post-Doctoral Researchers at CISPA Helmholtz Center for Information Security, Germany.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alcón, L.: On asteroidal sets in chordal graphs. Discret. Appl. Math. **164**, 482–491 (2014). https://doi.org/10.1016/j.dam.2013.04.019
2. Arvind, V., Nedela, R., Ponomarenko, I., Zeman, P.: Testing isomorphism of chordal graphs of bounded leafage is fixed-parameter tractable. *CoRR*, abs/2107.10689, (2021). arXiv:2107.10689
3. Balakrishnan, H., Rajaraman, A., Rangan, C.P.: Connected domination and steiner set on asteroidal triple-free graphs. In Frank K. H. A. D., Jörg-Rüdiger, S., Nicola, S., Sue W. (eds.) Algorithms and Data Structures, Third Workshop, WADS '93, Montréal, Canada, August 11-13, 1993, Proceedings, volume 709 of Lecture Notes in Computer Science, pp. 131–141. Springer (1993). https://doi.org/10.1007/3-540-57155-8_242
4. Barnetson, K.D., Burgess, A.C., Enright, J.A., Howell, J., Pike, D.A., Ryan, B.: The firebreak problem. Networks **77**(3), 372–382 (2021). https://doi.org/10.1002/net.21975
5. Belmonte, R., Kim, E.J., Lampis, M., Mitsou, V., Otachi, Y., Sikora, F.: Token sliding on split graphs. Theory Comput. Syst. **65**(4), 662–686 (2021). https://doi.org/10.1007/s00224-020-09967-8
6. Bergougnoux, B., Kanté, M.M.: More applications of the d-neighbor equivalence: acyclicity and connectivity constraints. SIAM J. Discret. Math. **35**(3), 1881–1926 (2021). https://doi.org/10.1137/20M1350571

7. Bergougnoux, B., Papadopoulos, C., Telle, J.A.: Node multiway cut and subset feedback vertex set on graphs of bounded mim-width. Algorithmica **84**(5), 1385–1417 (2022). https://doi.org/10.1007/s00453-022-00936-w
8. Bertossi, A.A.: Dominating sets for split and bipartite graphs. Inf. Process. Lett. **19**(1), 37–40 (1984). https://doi.org/10.1016/0020-0190(84)90126-1
9. Bousquet, N., Daligault, J., Thomassé, S.: Multicut is FPT. SIAM J. Comput. **47**(1), 166–207 (2018). https://doi.org/10.1137/140961808
10. Bui-Xuan, B.M., Telle, J.A., Vatshelle, M.: Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. Theor. Comput. Sci. **511**, 66–76 (2013). https://doi.org/10.1016/j.tcs.2013.01.009
11. Buneman, P.: A characterisation of rigid circuit graphs. Discret. Math. **9**(3), 205–212 (1974). https://doi.org/10.1016/0012-365X(74)90002-8
12. Cao, Y.: Linear recognition of almost interval graphs. In: Robert, K. (ed.) Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, pp. 1096–1115. SIAM, (2016). https://doi.org/10.1137/1.9781611974331.ch77
13. Chang, M.-S.: Efficient algorithms for the domination problems on interval and circular-arc graphs. SIAM J. Comput. **27**(6), 1671–1694 (1998). https://doi.org/10.1137/S0097539792238431
14. Chaplick, S., Stacho, J.: The vertex leafage of chordal graphs. Discret. Appl. Math. **168**, 14–25 (2014). https://doi.org/10.1016/j.dam.2012.12.006
15. Chitnis, R.H., Cygan, M., Hajiaghayi, M.T., Marx, D.: Directed subset feedback vertex set is fixed-parameter tractable. ACM Trans. Algorithms **11**(4), 28:1-28:28 (2015). https://doi.org/10.1145/2700209
16. Chitnis, R., Hajiaghayi, M.T., Marx, D.: Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. SIAM J. Comput. **42**(4), 1674–1696 (2013). https://doi.org/10.1137/12086217X
17. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015)
18. de Figueiredo, C. M., Lopes, R., de Melo, A. A., Silva, A.: Parameterized algorithms for steiner tree and dominating set: Bounding the leafage by the vertex leafage. In: Petra, M., Rahman, M. S., Slamin, (eds.) WALCOM: Algorithms and Computation - 16th International Conference and Workshops, WALCOM 2022, Jember, Indonesia, Proceedings, volume 13174 of Lecture Notes in Computer Science, pp. 251–262. Springer, (2022). https://doi.org/10.1007/978-3-030-96731-4_21
19. Diestel, R.: Graph Theory. volume 173 of Graduate texts in mathematics, 4th edn. Springer, Berlin (2012)
20. Drange, P.G., Dregi, M.S., van 't Hof, P.: On the computational complexity of vertex integrity and component order connectivity. Algorithmica **76**(4), 1181–1202 (2016). https://doi.org/10.1007/s00453-016-0127-x
21. Fomin, F.V., Golovach, P.A., Raymond, J.-F.: On the tractability of optimization problems on H-graphs. Algorithmica **82**(9), 2432–2473 (2020). https://doi.org/10.1007/s00453-020-00692-9
22. Fomin, F.V., Heggernes, P., Kratsch, D., Papadopoulos, C., Villanger, Y.: Enumerating minimal subset feedback vertex sets. Algorithmica **69**(1), 216–231 (2014). https://doi.org/10.1007/s00453-012-9731-6
23. Fomin, F.V., Kratsch, D., Woeginger, G.J.: Exact (exponential) algorithms for the dominating set problem. In: Hromkovic, J., Nagl, M., Westfechtel, B. (eds.) *Graph-Theoretic Concepts in Computer Science, 30th International Workshop, WG 2004, Bad Honnef, Germany, Revised Papers*, volume 3353 of *Lecture Notes in Computer Science*, pages 245–256. Springer, (2004). https://doi.org/10.1007/978-3-540-30559-0_21
24. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Can. J. Math. **8**, 399–404 (1956). https://doi.org/10.4153/CJM-1956-045-5
25. Gavril, F.: The intersection graphs of subtrees in tree are exactly the chordal graphs. J. Comb. Theory Ser. B (1974). https://doi.org/10.1016/0095-8956(74)90094-X
26. Gilmore, P.C., Hoffman, A.J.: A characterization of comparability graphs and of interval graphs. Can. J. Math. **16**, 539–548 (1964). https://doi.org/10.4153/CJM-1964-055-5
27. Golovach, P.A., Heggernes, P., van't Hof, P., Paul, C.: Hadwiger number of graphs with small chordality. SIAM J. Discret. Math. **29**(3), 1427–1451 (2015). https://doi.org/10.1137/140975279
28. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Elsevier, Amsterdam (2004)

29. Guo, J., Hüffner, F., Kenar, E., Niedermeier, R., Uhlmann, J.: Complexity and exact algorithms for vertex multicut in interval and bounded treewidth graphs. Eur. J. Oper. Res. **186**(2), 542–553 (2008). https://doi.org/10.1016/j.ejor.2007.02.014
30. Habib, M., Stacho, J.: Polynomial-time algorithm for the leafage of chordal graphs. In: Amos, F., Peter, S. (eds.) *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 290–300. Springer, (2009). https://doi.org/10.1007/978-3-642-04128-0_27
31. Habib, M., Stacho, J.: Reduced clique graphs of chordal graphs. Eur. J. Comb. **33**(5), 712–735 (2012). https://doi.org/10.1016/j.ejc.2011.09.031
32. Hochstättler, W., Hurink, J.L., Manthey, B., Paulusma, D., Peis, B., Still, G.: In memoriam walter kern. Discret. Appl. Math. **303**, 2–3 (2021). https://doi.org/10.1016/j.dam.2021.08.034
33. Ioannidou, K., Mertzios, G.B., Nikolopoulos, S.D.: The longest path problem has a polynomial solution on interval graphs. Algorithmica **61**(2), 320–341 (2011). https://doi.org/10.1007/s00453-010-9411-3
34. Kang, D.Y., Kwon, O.J., Strømme, T.J., Telle, J.A.: A width parameter useful for chordal and co-comparability graphs. Theor. Comput. Sci. **704**, 1–17 (2017). https://doi.org/10.1016/j.tcs.2017.09.006
35. Keil, J.M.: Finding Hamiltonian circuits in interval graphs. Inf. Process. Lett. **20**(4), 201–206 (1985). https://doi.org/10.1016/0020-0190(85)90050-X
36. Konstantinidis, A.L., Papadopoulos, C.: Cluster deletion on interval graphs and split related graphs. Algorithmica **83**(7), 2018–2046 (2021). https://doi.org/10.1007/s00453-021-00817-8
37. Kratsch, D.: Finding the minimum bandwidth of an interval graphs. Inf. Comput. **74**(2), 140–158 (1987). https://doi.org/10.1016/0890-5401(87)90028-9
38. Kratsch, D., Stewart, L.: Approximating bandwidth by mixing layouts of interval graphs. SIAM J. Discret. Math. **15**(4), 435–449 (2002). https://doi.org/10.1137/S0895480199359624
39. Lekkeikerker, C., Boland, J.: Representation of a finite graph by a set of intervals on the real line. Fundam. Math. **51**(1), 45–64 (1962)
40. Lin, I.-J., McKee, T.A., West, D.B.: The leafage of a chordal graph. Discuss. Math. Graph Theory **18**(1), 23–48 (1998). https://doi.org/10.7151/dmgt.1061
41. Lueker, G.S., Booth, K.S.: A linear time algorithm for deciding interval graph isomorphism. J. ACM **26**(2), 183–195 (1979). https://doi.org/10.1145/322123.322125
42. Marx, D.: Parameterized graph separation problems. Theor. Comput. Sci. **351**(3), 394–406 (2006). https://doi.org/10.1016/j.tcs.2005.10.007
43. Marx, D., Razgon, I.: Fixed-parameter tractability of multicut parameterized by the size of the cutset. SIAM J. Comput. **43**(2), 355–388 (2014). https://doi.org/10.1137/110855247
44. Misra, P., Panolan, F., Rai, A., Saurabh, S., Sharma, R.: Quick separation in chordal and split graphs. In: Javier, E., Daniel, K. (eds.) *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pp. 70:1–70:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. https://doi.org/10.4230/LIPIcs.MFCS.2020.70
45. Papadopoulos, C.: Restricted vertex multicut on permutation graphs. Discret. Appl. Math. **160**(12), 1791–1797 (2012). https://doi.org/10.1016/j.dam.2012.03.021
46. Papadopoulos, C., Tzimas, S.: Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs. Discret. Appl. Math. **258**, 204–221 (2019). https://doi.org/10.1016/j.dam.2018.11.017
47. Papadopoulos, C., haris, Tzimas, Spyridon: Computing a minimum subset feedback vertex set on chordal graphs parameterized by leafage. In Cristina Bazgan and Henning Fernau, editors, *Combinatorial Algorithms - 33rd International Workshop, IWOCA 2022, Trier, Germany, June 7-9, 2022, Proceedings*, volume 13270 of *Lecture Notes in Computer Science*, pp. 466–479. Springer, (2022). https://doi.org/10.1007/978-3-031-06678-8_34
48. Walter, J.R.: Representations of Rigid Cycle Graphs. Wayne State University, Detroit (1972)
49. White, K., Farber, M., Pulleyblank, W.R.: Steiner trees, connected domination and strongly chordal graphs. Networks **15**(1), 109–124 (1985). https://doi.org/10.1002/net.3230150109