

Feedback Vertex Set Inspired Kernel for Chordal Vertex Deletion

AKANKSHA AGRAWAL, Hungarian Academy of Sciences, Hungary

DANIEL LOKSHTANOV, University of Bergen, Norway

PRANABENDU MISRA, University of Bergen, Norway

SAKET SAURABH, Institute of Mathematical Sciences, India and University of Bergen, Norway

MEIRAV ZEHAVI, Ben-Gurion University, Israel

Given a graph G and a parameter k , the CHORDAL VERTEX DELETION (CVD) problem asks whether there exists a subset $U \subseteq V(G)$ of size at most k that hits all induced cycles of size at least 4. The existence of a polynomial kernel for CVD was a well-known open problem in the field of Parameterized Complexity. Recently, Jansen and Pilipczuk resolved this question affirmatively by designing a polynomial kernel for CVD of size $O(k^{161} \log^{58} k)$, and asked whether one can design a kernel of size $O(k^{10})$ [Jansen and Pilipczuk, SODA 2017]. While we do not completely resolve this question, we design a significantly smaller kernel of size $O(k^{12} \log^{10} k)$, inspired by the $O(k^2)$ -size kernel for FEEDBACK VERTEX SET [Thomassé, TALG 2010]. Furthermore, we introduce the notion of the independence degree of a vertex, which is our main conceptual contribution.

CCS Concepts: • **Theory of computation** → **Fixed parameter tractability**;

Additional Key Words and Phrases: chordal vertex deletion, chordal graph, parameterized complexity, kernelization

ACM Reference Format:

Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. 2010. Feedback Vertex Set Inspired Kernel for Chordal Vertex Deletion. *ACM Trans. Web* 9, 4, Article 39 (March 2010), 28 pages. <https://doi.org/0000001.0000001>

1 INTRODUCTION

Data reduction techniques are widely applied to deal with computationally hard problems in real world applications. It has been a long-standing challenge to formally express the efficiency and accuracy of these “pre-processing” procedures. The framework of parameterized complexity turns out to be particularly suitable for a mathematical analysis of pre-processing heuristics. Formally, in parameterized complexity each problem instance is accompanied by a parameter k , and we say that a parameterized problem is *fixed-parameter tractable* (FPT) if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is a computable

A preliminary version of this paper appeared in the proceedings of 28th ACM-SIAM Symposium on Discrete Algorithms (SODA 2017). The research leading to these results received funding from the European Research Council under the European Unions Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 306992.

Authors' addresses: Akanksha Agrawal, Hungarian Academy of Sciences, Budapest, Hungary, akanksha@sztaki.mta.hu; Daniel Lokshtanov, University of Bergen, Bergen, Norway, daniello@ii.uib.no; Pranabendu Misra, University of Bergen, Bergen, Norway, pranabendu.m@gmail.com; Saket Saurabh, Institute of Mathematical Sciences, Chennai, India, University of Bergen, Bergen, Norway, saket@imsc.res.in; Meirav Zehavi, Ben-Gurion University, Beersheba, Israel, meiravze@bgu.ac.il.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2009 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

1559-1131/2010/3-ART39 \$15.00

<https://doi.org/0000001.0000001>

function of the parameter k alone. *Kernelization* is the subarea of parameterized complexity that deals with the mathematical analysis of pre-processing heuristics. A parameterized problem is said to admit a *polynomial kernel* if there is a polynomial-time algorithm (the degree of polynomial is independent of the parameter k), called a *kernelization algorithm*, that reduces the input instance down to an instance whose size is bounded by a polynomial $p(k)$ in k , while preserving the answer. This reduced instance is called a $p(k)$ -*kernel* for the problem. Observe that if a problem has a kernelization algorithm, then it also has an FPT algorithm.

Kernelization appears to be an interesting computational approach not only from a theoretical perspective, but also from a practical perspective. There are many real-world applications where even very simple preprocessing can be surprisingly effective, leading to significant size-reduction of the input. Kernelization is a natural tool for measuring the quality of existing preprocessing rules proposed for specific problems as well as for designing new powerful such rules. The most fundamental question in the field of kernelization is:

Let Π be parameterized problem that admits an FPT algorithm. Then, does Π admit a polynomial kernel?

In recent times, the study of kernelization, centred on the above question, has been one of the main areas of research in parameterized complexity, yielding many new important contributions to theory. These include general results showing that certain classes of parameterized problems have polynomial kernels, as well as other results that utilize advanced techniques from algebra, matroid theory and topology for data reduction [4, 6, 17, 18, 22, 25, 28, 29, 31, 32, 39, 40]. The development of a framework for ruling out polynomial kernels under certain complexity-theoretic assumptions [5, 7, 14, 20] has added a new dimension to the area, and strengthened its connections to classical complexity theory. We refer to the following surveys [30, 34] and the corresponding chapters in the books [12, 15, 16, 38], for a detailed introduction to the field of kernelization.

An important class of problems, that has led to the development of many upper bound tools and techniques in kernelization, is the class of *parameterized graph deletion problems*. A typical problem of this class is associated with a family of graphs, \mathcal{F} , such as edgeless graphs, forests, cluster graphs, chordal graphs, interval graphs, bipartite graphs, split graphs or planar graphs. The deletion problem corresponding to \mathcal{F} is formally stated as follows.

\mathcal{F} -VERTEX (EDGE) DELETION

Parameter: k

Input: An undirected graph G and a non-negative integer k .

Question: Does there exist $S \subseteq V(G)$ (or $S \subseteq E(G)$) such that $|S| \leq k$ and $G \setminus S$ is in \mathcal{F} ?

Graph deletion problems are also among the most basic problems in graph theory and graph algorithms. Most of these problems are NP-complete [33, 41], and thus they were subject to intense study in various algorithmic paradigms to cope with their intractability [17, 21, 35, 37]. These include, considering a restricted class of inputs, approximation algorithms, parameterized complexity and kernelization.

Some of the most well known results in kernelization are polynomial kernels for graph deletion problems such as FEEDBACK VERTEX SET [40], ODD CYCLE TRANSVERSAL [31, 32], VERTEX COVER [2, 11], PLANAR- \mathcal{F} -DELETION [17], and TREEDEPTH- η -DELETION [22]. A common thread among all these problems, with the exception of ODD CYCLE TRANSVERSAL, is that the corresponding family \mathcal{F} can be characterized by a finite set of forbidden minors that include at least one connected planar graph. It is known that, if \mathcal{F} can be characterized by a finite set of forbidden induced subgraphs, then the corresponding \mathcal{F} -VERTEX DELETION problem immediately admits an FPT algorithm as well as polynomial sized kernel because of its connection to d -HITTING SET [1]. However, if \mathcal{F} is characterized by an *infinite* set of forbidden induced subgraphs, which is the case when \mathcal{F} is

the class of chordal graphs, chordal bipartite graphs, interval graphs, proper interval graphs and permutation graphs, our understanding of these problems in the realm of parameterized complexity and kernelization, is still at a nascent stage. While CHORDAL VERTEX DELETION (CVD) was known to be FPT for some time [10, 36], the parameterized complexity of INTERVAL VERTEX DELETION was settled only recently [8, 9]. The parameterized complexity of PERMUTATION VERTEX DELETION and CHORDAL BIPARTITE VERTEX DELETION is still unknown. Coming to the question of polynomial kernels for these problems, the situation is even more grimmer. Until recently, the only known result was a polynomial kernel for PROPER INTERVAL VERTEX DELETION: Fomin et al. [19] obtained a $O(k^{53})$ sized polynomial kernel for PROPER INTERVAL VERTEX DELETION, which has recently been improved to $O(k^4)$ [27]. A dearth of further results in this area has led to the questions of kernelization complexity of CHORDAL VERTEX DELETION and INTERVAL VERTEX DELETION becoming prominent open problems [10, 13, 19, 24, 36].

A recent work [26] has resolved one of these open questions. They showed that CVD admits a polynomial kernel of size $O(k^{161} \log^{58} k)$, and further posed an open question:

Does CVD admit a kernel of size $O(k^{10})$?

While we do not completely resolve this question, we design a significantly smaller kernel. In particular, we obtain the following result.

THEOREM 1.1. *CVD admits a polynomial kernel of size $O(k^{12} \log^{10} k)$.*

Our Methods. Our result is inspired by the $O(k^2)$ -size kernel for FEEDBACK VERTEX SET (FVS) (checking whether there exists a k sized vertex subset that intersects all cycles), designed by [40]. The kernel for FVS consists of the two following steps.

- (1) Reduce the maximum degree of the graph by using matching based tools (in particular expansion lemma). That is, upper bound the maximum degree Δ of the graph by $O(k)$.
- (2) When the graph has maximum degree Δ , one can show that if a graph has minimum degree at least 3 (which can be easily achieved for FVS) then any minimum feedback vertex set has size $\Omega(n/\Delta)$. This together with an upper bound on Δ implies $O(k^2)$ -size kernel.

Let us now look at the CVD problem. Here, our objective is to check whether there exists a k -sized vertex subset S such that $G \setminus S$ is a chordal graph, i.e., a graph without induced cycles of length at least 4 (also called chordless cycles). Thus, FVS is about intersecting all cycles and CVD is about intersecting all chordless cycles. Unfortunately, this apparent similarity stops here! Nevertheless, we are still able to exploit ideas used in the $O(k^2)$ -size kernel for FVS. Towards this, we define the notion of *independence degree* of vertices and graphs. Roughly speaking, the independence degree of a vertex v is the size of a maximum independent set in its neighborhood ($G[N(v)]$). The study of this notion is our main conceptual contribution.

As a first step we bound the independence degree of every vertex by $k^{O(1)}$ – this is similar to the first step of the kernel for FVS. Once we have bounded the independence degree of a graph, we obtain an approximate solution M (also called modulator) and analyze the graph $G \setminus M$. The bound on the independence degree immediately implies that the number of leaves, vertices of degree at least three and the number of maximal degree two paths in the clique forest of $G \setminus M$ is bounded by $k^{O(1)}$. Then, using ideas similar to those used in [36] to bound the size of a maximal clique while designing the first FPT algorithm for CVD, we reduce the size of each maximal clique in $G \setminus M$ to $k^{O(1)}$. Finally, we use structural analysis to bound the size of each maximal degree two path, which includes the design of a reduction rule that computes a family of minimum cuts, and thus we obtain our final kernel. We believe that the notion of independent degree is likely to be useful for designing algorithms for other graph modification problems. Not only does it lead to a significant improvement, once it is bounded, it greatly simplifies the analysis of the resulting instance.

Finally, after we obtain a kernel, we show that by rerunning our entire kernelization procedure once, we can actually reduce the size of the kernel. Since we rely on an $O(\log^2 n)$ -approximation algorithm, when we call our kernelization procedure for the first time, we work with an approximate solution of size $O(k^3 \log^2 k)$; indeed, it can be assumed that $\log n < k \log k$, else the $2^{O(k \log k)} \cdot n^{O(1)}$ -time algorithm for CVD by [10] solves the input instance in polynomial time. However, once we have our first kernel, it holds that $n = O(k^{22} \log^{12} k)$. At this point, if we reuse the approximation algorithm, we obtain an approximate solution of size $O(k \log^2 k)$. The size of our kernel depends on the size of the approximate solution; in particular, having an approximate solution of size $O(k \log^2 k)$ allows us to obtain a kernel of size $O(k^{12} \log^{10} k)$.

2 PRELIMINARIES

For a positive integer k , we use $[k]$ as a shorthand for $\{1, 2, \dots, k\}$. Given a function $f : A \rightarrow B$ and a subset $A' \subseteq A$, we let $f|_{A'}$ denote the function f restricted to the domain A' .

Parameterized Complexity. In Parameterized Complexity each problem instance is accompanied by a parameter k . A central notion in this field is the one of *fixed-parameter tractability (FPT)*. This means, for a given instance (I, k) , solvability in time $f(k)|I|^{O(1)}$ where f is some computable function of k . A parameterized problem is said to admit a *polynomial kernel* if there is a polynomial-time algorithm (the degree of polynomial is independent of the parameter k), called a *kernelization algorithm*, that reduces the input instance down to an equivalent instance whose size is bounded by a polynomial $p(k)$ in k . Here, two instances are equivalent if one of them is a yes-instance if and only if the other one is a yes-instance. The reduced instance is called a $p(k)$ -kernel for the problem. For a detailed introduction to the field of kernelization, we refer to the following surveys [30, 34] and the corresponding chapters in the books [12, 15, 16, 38].

Kernelization algorithms often rely on the design of *reduction rules*. The rules are numbered, and each rule consists of a condition and an action. We always apply the first rule whose condition is true. Given a problem instance (I, k) , the rule computes (in polynomial time) an instance (I', k') of the same problem where $k' \leq k$. Typically, $|I'| < |I|$, where if this is not the case, it should be argued why the rule can be applied only polynomially many times. We say that the rule is *safe* if the instances (I, k) and (I', k') are equivalent.

The standard definition of an approximation algorithm for a parameterized problem is as follows. We say that a parameterized minimization problem Π admits a *c-factor approximation algorithm* \mathcal{A} , if given an instance (I, k) of Π , in polynomial time \mathcal{A} returns either (i) the answer No, in which case there is no solution of value (size) at most k , or (ii) a solution of value (size) at most $c \cdot k$.

Graphs. Given a graph G , we let $V(G)$ and $E(G)$ denote its vertex-set and edge-set, respectively. In this paper, we only consider undirected graphs. We let $n = |V(G)|$ denote the number of vertices in the graph G , where G will be clear from context. The *open neighborhood*, or simply the *neighborhood*, of a vertex $v \in V(G)$ is defined as $N_G(v) = \{w \mid \{v, w\} \in E(G)\}$. The *closed neighborhood* of v is defined as $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of v is defined as $d_G(v) = |N_G(v)|$. We can extend the definition of neighborhood of a vertex to a set of vertices as follows. Given a subset $U \subseteq V(G)$, $N_G(U) = \bigcup_{u \in U} N_G(u)$ and $N_G[U] = \bigcup_{u \in U} N_G[u]$. The *induced subgraph* $G[U]$ is the graph with vertex-set U and edge-set $\{\{u, u'\} \mid u, u' \in U, \text{ and } \{u, u'\} \in E(G)\}$. Moreover, we define $G \setminus U$ as the induced subgraph $G[V(G) \setminus U]$. We omit subscripts when the graph G is clear from context. An *independent set* in G is a set of vertices such that there is no edge between any pair of vertices in this set. The *independence number* of G , denoted by $\alpha(G)$, is defined as the cardinality of the largest independent set in G . A *clique* in G is a set of vertices such that there is an edge between every pair of vertices in this set.

A path P in G is a subgraph of G where $V(P) = \{x_1, x_2, \dots, x_\ell\} \subseteq V(G)$ and

$$E(P) = \{\{x_1, x_2\}, \{x_2, x_3\}, \dots, \{x_{\ell-1}, x_\ell\}\} \subseteq E(G)$$

for some $\ell \in [n]$. The vertices x_1 and x_ℓ are called *endpoints* of the path P and the remaining vertices in $V(P)$ are called *internal vertices* of P . We also say that P is a path between x_1 and x_ℓ . A cycle C in G is a subgraph of G where $V(C) = \{x_1, x_2, \dots, x_\ell\} \subseteq V(G)$ and $E(C) = \{\{x_1, x_2\}, \{x_2, x_3\}, \dots, \{x_{\ell-1}, x_\ell\}, \{x_\ell, x_1\}\} \subseteq E(G)$, i.e., it is a path with an additional edge between x_1 and x_ℓ . Let P be a path in the graph G on at least three vertices. We say that $\{u, v\} \in E(G)$ is a *chord* of P if $u, v \in V(P)$ but $\{u, v\} \notin E(P)$. Similarly, for a cycle C on at least four vertices, $\{u, v\} \in E(G)$ is a chord of C if $u, v \in V(C)$ but $\{u, v\} \notin E(C)$. A path P or cycle C is *chordless* if it has no chords. Let us note that any chordless cycle has length at least 4.

The graph G is *connected* if there is a path between every pair of vertices, otherwise G is *disconnected*. A connected graph without any cycles is a *tree*, and a collection of trees is a *forest*. A maximal connected subgraph of G is called a *connected component* of G .

Forest Decompositions. A forest decomposition of a graph G is a pair (F, β) where F is forest, and $\beta : V(T) \rightarrow 2^{V(G)}$ is a function that satisfies the following,

(i) $\bigcup_{v \in V(F)} \beta(v) = V(G)$,

(ii) for any edge $\{v, u\} \in E(G)$ there is a node $w \in V(F)$ such that $v, u \in \beta(w)$,

(iii) and for any $v \in V(G)$, the collection of nodes $T_v = \{u \in V(F) \mid v \in \beta(u)\}$ is a subtree of F .

For $v \in V(F)$, we call $\beta(v)$ the *bag* of v , and for the sake of clarity of presentation, we sometimes use v and $\beta(v)$ interchangeably. We refer to the vertices in $V(F)$ as nodes. A *tree decomposition* is a forest decomposition where F is a tree.

Chordal Graphs. A graph G is a *chordal graph* if it has no chordless cycle as an induced subgraph, i.e., every cycle of length at least four has a chord. A *clique forest* of G is a forest decomposition of G where every bag is a maximal clique. We further insist that every bag of the clique forest is distinct. The following lemma shows that the class of chordal graphs is exactly the class of graphs which have a clique forest.

LEMMA 2.1 (THEOREM 4.8, [23]). *A graph G is a chordal graph if and only if G has a clique forest.*

Given a subset $U \subseteq V(G)$, we say that U *hits* a chordless cycle C in G if $U \cap V(C) \neq \emptyset$. Observe that if U hits every chordless cycle of G , then $G \setminus U$ is a chordal graph. Given a $v \in V(G)$, we say that a vertex-set $B \subseteq V(G) \setminus \{v\}$ is a *v -blocker* if B hits every chordless cycle in G . Observe that the set B must *not* contain the vertex v .

The CHORDAL VERTEX DELETION (CVD) problem is defined as follows.

CHORDAL VERTEX DELETION (CVD)

Parameter: k

Input: An undirected graph G and a non-negative integer k .

Question: Does there exist a subset $S \subseteq V(G)$ such that $|S| \leq k$ and $G \setminus S$ is a chordal graph?

For purposes of approximation, we also formulate this problem as an optimization problem.

CHORDAL VERTEX DELETION (CVD)

Input: An undirected graph G .

Question: What is the minimum cardinality of a subset $S \subseteq V(G)$ such that $G \setminus S$ is a chordal graph?

The Expansion Lemma. Let c be a positive integer. A *c -star* is a graph on $c + 1$ vertices where

one vertex, called the center, has degree c , and all other vertices are adjacent to the center and have degree one. A *bipartite graph* is a graph whose vertex-set can be partitioned into two independent sets. Such a partition of the vertex-set is called a *bipartition* of the graph. Let G be a bipartite graph with bipartition (A, B) . A subset of edges $M \subseteq E(G)$ is called *c -expansion of A into B* if

- (i) every vertex of A is incident to exactly c edges of M ,
- (ii) and M saturates exactly $c|A|$ vertices in B .

Note that a c -expansion saturates all vertices of A , and for each $u \in A$ the set of edges in M incident on u form a c -star. The following lemma allows us to compute a c -expansion in a bipartite graph. It captures a certain property of neighborhood sets which is very useful for designing kernelization algorithms.

LEMMA 2.2 ([12, 40]). *Let G be a bipartite graph with bipartition (A, B) such that there are no isolated vertices in B . Let c be a positive integer such that $|B| \geq c|A|$. Then, there are non-empty subsets $X \subseteq A$ and $Y \subseteq B$ such that*

- *there is a c -expansion from X into Y ,*
- *and there is no vertex in Y that has a neighbor in $A \setminus X$, i.e. $N_G(Y) = X$.*

Further, the sets X and Y can be computed in polynomial time.

3 KERNELIZATION

In this section we prove Theorem 1.1. First, in Section 3.1, we briefly state results relating to approximate solutions for CVD, which will be relevant to following subsections. In Section 3.2 we address annotations that will be added to the input instance. Next, in Section 3.3, we introduce the notion of the *independent degree* of a vertex, which lies at the heart of the design of our kernelization algorithm. We carefully examine the independent degrees of vertices in our graphs, and show how these degrees can be bounded by a small polynomial in k . In Section 3.4, we consider the clique forest of the graph obtained by removing (from the input graph) the vertices of an approximate solution. In particular, we demonstrate the usefulness of our notion of an independent degree of a vertex – having bounded the independent degree of each vertex, we show that the number of leaves in the clique forest can be bounded in a simple and elegant manner. We also efficiently bound the size of a maximal clique. Then, in Section 3.5, we turn to bound the length of degree-2 paths in the clique forest. This subsection is quite technical, and its proofs are based on insights into the structure of chordal graphs and their clique forests. In particular, we use a reduction rule which computes a collection of minimum cuts rather than one minimum cut which overall allows us to capture the complexity of a degree-2 path using only few vertices. In Section 3.6, we remove annotations introduced in preceding sections. Next, in Section 3.7, we bound the size of our kernel. Finally, in Section 3.8, we show that an alternating application of approximation and kernelization can improve the performance of our kernelization algorithm.

3.1 Approximation

Observe that it can be assumed that $\log n < k \log k$, else the $2^{O(k \log k)} \cdot n^{O(1)}$ -time algorithm for CVD by [10] runs in time $2^{O(k \log k)} \cdot n^{O(1)} < 2^{O(\log n)} \cdot n^{O(1)}$, and hence solves the input instance in polynomial time. Thus, since CVD admits an $O(\log^2 n)$ -factor approximation algorithm [3], we obtain the following result.

LEMMA 3.1. *CVD admits an $O(k^2 \log^2 k)$ -factor approximation algorithm.*

Throughout Section 3, we let APPROX denote a polynomial-time algorithm for CVD that returns approximate solutions of size $f(\text{opt})$ for some function f . Initially, it will denote the algorithm given by Lemma 3.1. Given an instance of CVD, we say that an approximate solution D is *redundant*

if for every vertex $v \in D$, $D \setminus \{v\}$ is also an approximate solution, that is, $G \setminus (D \setminus \{v\})$ is a chordal graph. Jansen and Pilipczuk [26] showed the following result.

COROLLARY 3.2 ([26]). *Suppose that CHORDAL VERTEX DELETION admits an approximation solution of size $f(\text{opt})$, for some function f , in polynomial time. Then given an instance (G, k) , in polynomial time, one can find either a vertex that is contained in every solution of size k or a redundant approximate solution of size $O(k \cdot f(k))$.*

Next, we fix an instance (G, k) of CVD. By relying on APPROX and Corollary 3.2, we may assume that we have a vertex-set $\tilde{D} \subseteq V(G)$ that is an approximate solution of size $f(k)$ and a vertex-set $D \subseteq V(G)$ that is a redundant approximate solution of size $c \cdot k \cdot f(k)$ for some constant c independent of the input.

In the following subsection, we will also need to strengthen approximate solutions to be v -blockers for some vertices $v \in V(G)$. To this end, we will rely on the following result.

LEMMA 3.3. *Given a vertex $v \in V(G)$, one can find (in polynomial time) either a vertex contained in every solution of size at most k or an approximate solution of size $f(k)$ that is a v -blocker.*

PROOF. Fix a vertex $v = v_0 \in V(G)$. We define the graph G' by setting $V(G') = V(G) \cup \{v_1, v_2, \dots, v_{f(k)}\}$, where $v_1, v_2, \dots, v_{f(k)}$ are new vertices, and $E(G') = E(G) \cup \{\{u, v_i\} : \{u, v\} \in E(G), i \in [f(k)]\} \cup \{(v_i, v_j) : i \neq j \in \{0\} \cup [f(k)]\}$. In other words, G' is the graph G to which we add $f(k)$ copies of the vertex v , which form a clique amongst themselves and v .

We call the algorithm APPROX on G' to obtain an approximate solution S . Since $G'[\{v_0, v_1, \dots, v_{f(k)}\}]$ is a clique, any chordless cycle in G' contains at most one vertex from $\{v_0, v_1, \dots, v_{f(k)}\}$. In particular, for any chordless cycle C in G' , by replacing v_i by v_0 (in case v_i belongs to C), we obtain a chordless cycle in G . Therefore, if the instance (G, k) admits a solution of size at most k that does not contain v , then this solution is also a solution for the instance (G', k) , which implies that the size of S should be at most $f(k)$. Thus, we can next assume that $|S| \leq f(k)$, else we conclude that the vertex v is contained in every solution of size at most k . Since the vertices $v_0, v_1, v_2, \dots, v_{f(k)}$ have the same neighbor-set, if $\{v_0, v_1, v_2, \dots, v_{f(k)}\} \cap S \neq \emptyset$, we can assume that $\{v_0, v_1, v_2, \dots, v_{f(k)}\} \subseteq S$, since otherwise we can take $S \setminus \{v_0, v_1, v_2, \dots, v_{f(k)}\}$ as our approximate solution S . Since we assume that $|S| \leq f(k)$, we deduce that $\{v_0, v_1, v_2, \dots, v_{f(k)}\} \cap S = \emptyset$. In particular, we have that $v \notin S$ and therefore S is a v -blocker. \square

In light of Lemma 3.3, we may next assume that for every vertex $v \in V(G)$, we have a vertex-set B_v that is both a v -blocker and an approximate solution of size $f(k)$.

3.2 Irrelevant and Mandatory Edges

During the execution of our kernelization algorithm, we mark some edges in $E(G)$ as *irrelevant edges*. At the beginning of its execution, all of the edges in $E(G)$ are assumed to be relevant edges. When we mark an edge as an irrelevant edge, we prove that any solution that hits all of the chordless cycles in G that contain only relevant edges also hits all of the chordless cycles in G that contain the irrelevant edge. In other words, we prove that we can safely ignore chordless cycles that contain at least one irrelevant edge. Observe that we cannot simply remove irrelevant edges from $E(G)$ since this operation may introduce new chordless cycles in G . Instead we maintain a set E_I , which contains the edges marked as irrelevant.

We also mark some edges in $E(G)$ as *mandatory edges*. We will ensure that at least one endpoint of a mandatory edge is present in any solution of size at most k . We let E_M denote the set of mandatory edges.

In some situations, we identify a pair of non-adjacent vertices such that any solution of size at most k must contain at least one of them. Then, we add an edge between the vertices of the

pair, and mark this edge as a mandatory edge. The correctness of this operation follows from the observation that any chordless cycle affected by the addition of this edge contains both vertices of the pair, and since the edge is marked as a mandatory edge, we ensure this chordless cycle will be hit although it may no longer exist. Formally, we have the following reduction rule.

REDUCTION RULE 1. *Given two non-adjacent vertices in G , v and u , such that at least one of them belongs to any solution of size at most k , insert the edge $\{v, u\}$ into both $E(G)$ and E_M .*

Hence from now onwards our instance is of the form (G, k, E_I, E_M) , and during the execution of our kernelization algorithm, we will update the sets E_I and E_M . In Section 3.6, we show that we can unmark the edges in $E_I \cup E_M$, obtaining an ordinary instance of CVD. For the sake of simplicity, when E_I and E_M are clear from context, we omit them.

The Number of Mandatory Edges. If a vertex v is incident to at least $k + 1$ mandatory edges, it must belong to any solution of size at most k . Therefore, we may safely apply the following reduction rule.

REDUCTION RULE 2. *If there exists a vertex v incident to at least $k + 1$ mandatory edges, remove v from G and decrement k by 1.*

After exhaustively applying the above reduction rule we have the following lemma.

LEMMA 3.4. *If $|E_M| > k^2$ then the input instance is a no-instance.*

PROOF. After the exhaustive application of Reduction Rule 2, any vertex incident to a mandatory edge is incident to at most k such edges. Therefore, any set of at most k vertices from $V(G)$ may intersect at most k^2 mandatory edge. Since every solution of size at most k must intersect every mandatory edge, we deduce that if $|E_M| > k^2$, the input instance is a no-instance. \square

Thus, we will next assume that $|E_M| \leq k^2$ (else Reduction Rule 2 applies). In particular, whenever we find a new mandatory edge, we apply the above reduction rule. Moreover, we let D' denote the set $D \cup \tilde{D}$ to which we add every vertex that is an endpoint of an edge in E_M (recall that \tilde{D} is our approximate solution of size at most $f(k)$ and D is our redundant approximate solution of size $O(k \cdot f(k))$). Observe that by adding vertices to a redundant approximate solution, it remains a redundant approximate solution, and therefore D' or any other superset of D is such a solution. Furthermore, we will also update D' whenever the set of mandatory edges E_M is updated.

3.3 Independent Degree

Given a vertex $v \in V(G)$, we use the notation $N_G^R(v)$ to refer to the set consisting of each vertex $u \in N_G(v)$ such that $\{v, u\}$ does not belong to $E_I \cup E_M$. We remark that in this subsection we identify and mark some edges as irrelevant edges.

Independent Degrees. We start by introducing the notion of the independent degree of vertices and graphs.

Definition 3.5. Given a vertex $v \in V(G)$, the *independent degree of v* , denoted by $d_G^I(v)$, is the size of a maximum independent set in the graph $G[N_G^R(v)]$. The *independent degree of G* , denoted by Δ_G^I , is the maximum independent degree of a vertex in $V(G)$.

Fix $\Delta = (k+3)f(k)$. The objective of this subsection is to investigate the notion of an independent degree, ultimately proving the following result.

LEMMA 3.6. *One can construct (in polynomial time) an instance (G', k', E_I', E_R') of CVD that is equivalent to the input instance (G, k, E_I, E_R) and such that both $k' \leq k$ and $\Delta_{G'}^I \leq \Delta$.*

To this end, we may assume that we are given a (fixed) vertex $v \in V(G)$ such that $d_G^I(v) > \Delta$. Hereafter, in this subsection all the sets/graphs that are defined will be with respect to the fixed vertex v . We say that an instance (G', k', E'_I, E'_M) of CVD is *better* than the input instance (G, k, E_I, E_M) if $k' \leq k$, $V(G') = V(G)$, $E_I \subseteq E'_I$, $E_M \subseteq E'_M$, $d_{G'}^I(v) \leq \Delta$ and for all $u \in V(G')$, $d_{G'}^I(u) \leq d_G^I(u)$. To prove the correctness of Lemma 3.6, it is sufficient to prove the correctness of the following lemma.

LEMMA 3.7. *We can construct (in polynomial time) an instance (G', k', E'_I, E'_M) of CVD that is better than the input instance (G, k, E_I, E_M) .*

Indeed, to prove Lemma 3.6, one can repeatedly apply the operation given by Lemma 3.7 in the context of every vertex $u \in V(G)$ such that $d_G^I(u) > \Delta$. We start with a simple result concerning independent degrees.

LEMMA 3.8. *Let $u \in V(G)$ be a vertex such that $d_G^I(u) \geq |B_u|$. Then, one can find (in polynomial time) an independent set in $G[N_G^R(u) \setminus B_u]$ of size at least $d_G^I(u) - |B_u|$.*

PROOF. Since B_u is a solution, $G \setminus B_u$ is a chordal graph. In particular, $G[N_G^R(u) \setminus B_u]$ is a chordal graph. Since chordal graphs are perfect, MAXIMUM INDEPENDENT SET in chordal graphs is solvable in polynomial time [23]. This means that we can find a maximum independent set in $G[N_G^R(u) \setminus B_u]$ in polynomial time. Since the size of a maximum independent set in $G[N_G^R(u)]$ is at least $d_G^I(u)$, the size of a maximum independent set in $G[N_G^R(u) \setminus B_u]$ is at least $d_G^I(u) - |B_u|$, which concludes the correctness of the lemma. \square

Recall that for any vertex $u \in V(G)$, $|B_u| \leq f(k)$. Thus, we have the following corollary.

COROLLARY 3.9. *One can find (in polynomial time) an independent set in $G[N_G^R(v) \setminus B_v]$ of size at least $\Delta - f(k)$.*

We let I denote the independent set given by Corollary 3.9.

Independent Components. Let $X = N_G(v) \setminus (B_v \cup I)$ denote the neighbor-set of v from which we remove the vertices of the v -blocker B_v and of the independent set I . We also let $H = G \setminus (\{v\} \cup B_v \cup X)$ denote the graph obtained by removing (from G) the vertex v , the v -blocker B_v and any neighbor of v that does not belong to the independent set I . We define the set of independent components of v as the set consisting of each connected component of H that contains at least one vertex from I , and denote this set by \mathcal{A} .

For the set \mathcal{A} , we prove the following lemmata.

LEMMA 3.10. *Each connected component $A \in \mathcal{A}$ contains exactly one vertex from I and no other vertex from $N_G(v)$.*

PROOF. The graph H does not contain any vertex from $N_G(v) \setminus I$, and therefore we only need to prove the first part of the statement of the lemma. Fix a connected component $A \in \mathcal{A}$. By the definition of \mathcal{A} , it is only necessary to prove that A cannot contain (at least) two vertices from I . Suppose, by way of contradiction, that it contains two such vertices, u and w . Let P denote the shortest path in A that connects u and w . Since I is an independent set, this path contains at least two edges. Therefore, together with the vertex v , the path P forms a chordless cycle. However, this chordless cycle contains no vertex from B_v , which contradicts the fact that B_v is an approximate solution. \square

By Lemma 3.10, for each connected component $A \in \mathcal{A}$ we can let $z(A) \in I$ denote the unique neighbor of v in A . In fact, Corollary 3.9 and Lemma 3.10 imply that $\Delta - f(k) \leq |\mathcal{A}|$.

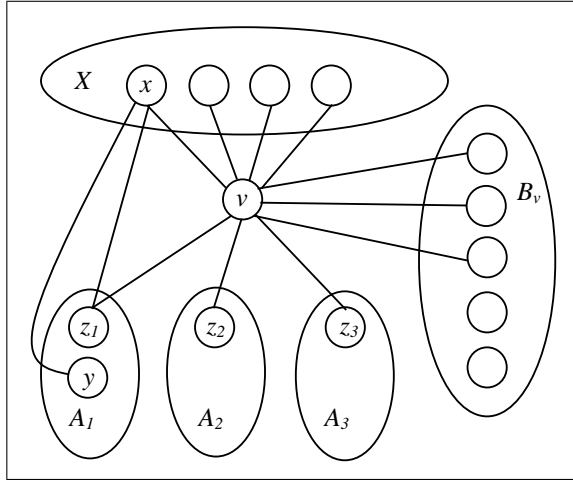


Fig. 1. The relations between v , B_v , X and \mathcal{A} .

LEMMA 3.11. *Every vertex $x \in X$ that is adjacent (in G) to some vertex $y \in V(A)$, where $A \in \mathcal{A}$, is also adjacent (in G) to the vertex $z(A)$.*

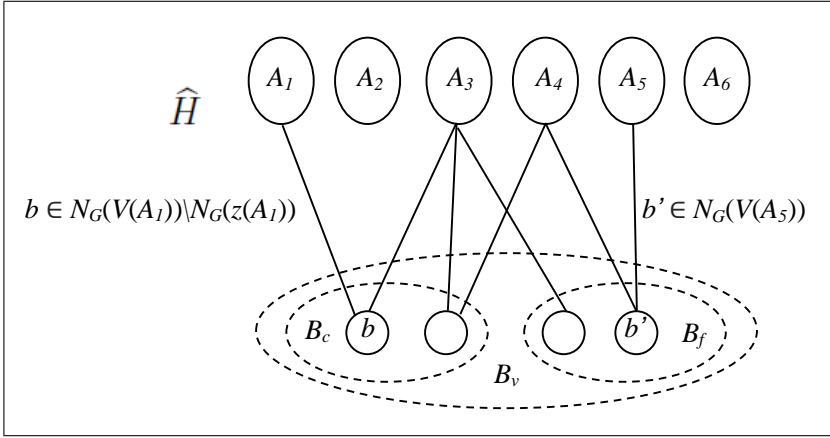
PROOF. Let $x \in X$ be a vertex that is adjacent (in G) to some vertex $y \in V(A)$. Suppose, by way of contradiction, that $\{x, z(A)\} \notin E(G)$. Then, $y \neq z(A)$. Let P denote a shortest path in $G[V(A) \cup \{v\}]$ that connects v and y . Let y' be the vertex on P that is closest to v , excluding v itself, among the neighbors of x . Moreover, let P' denote the subpath of P between v and y' . By Lemma 3.10, P' must contain the edge $\{v, z(A)\}$, and in particular P' contains at least two edges. By our choice of y' , the only neighbors of x on this path are v and y' . Moreover, as this path is a shortest path from v to y' in $G[V(A) \cup \{v\}]$, we derive that together with x , it forms a chordless cycle that includes v . Therefore, $G[V(P) \cup \{x\}]$ contains a chordless cycle that includes v . However, this chordless cycle contains no vertex from B_v , which contradicts the fact that B_v is an approximate solution. \square

An illustration of the relations between v , B_v , X and \mathcal{A} is given in Fig. 1.

The Bipartite Graph \widehat{H} . To decrease $d_G^l(v)$, we will consider the bipartite graph \widehat{H} , which is defined as follows. We define the vertex-set of \widehat{H} by $V(\widehat{H}) = \mathcal{A} \cup B_v$. In this context, we mean that each connected component $A \in \mathcal{A}$ is represented by a vertex in $V(\widehat{H})$, and for the sake of simplicity, we use the symbol A also to refer to this vertex. We partition B_v into two sets, B_c and B_f , where B_c contains the vertices in B_v that are adjacent (in G) to v , while B_f contains the remaining vertices in B_v . Here, the letters c and f stand for “close” and “far”. Having this partition, we define the edge-set of \widehat{H} as follows. For every vertex $b \in B_c$ and connected component $A \in \mathcal{A}$ such that $b \in N_G(V(A)) \setminus N_G(z(A))$ (i.e., b is a neighbor of some vertex in A but not of the vertex $z(A)$), insert the edge $\{b, A\}$ into $E(\widehat{H})$. Moreover, for every vertex $b \in B_f$ and connected component $A \in \mathcal{A}$ such that $b \in N_G(V(A))$, insert the edge $\{b, A\}$ into $E(\widehat{H})$. An illustration of the bipartite graph \widehat{H} is given in Fig. 2. The motivation behind its definition lies at the following lemma.

LEMMA 3.12. *The bipartite graph \widehat{H} satisfies the following properties.*

- (1) *Suppose that we are given an edge $\{b, A\} \in E(\widehat{H})$ such that $b \in B_c$ and $A \in \mathcal{A}$. Then, the graph G has a chordless cycle defined by the edges $\{v, b\}$ and $\{v, z(A)\}$ and a path in A .*

Fig. 2. The bipartite graph \widehat{H} .

(2) Suppose that we are given edges $\{b, A\}, \{b, A'\} \in E(\widehat{H})$ such that $b \in B_f$ and $A, A' \in \mathcal{A}$. Then, the graph G has a chordless cycle defined by the edges $\{v, z(A)\}, \{v, z(A')\}$, the vertex b and paths in A and A' .

PROOF. To prove the first item, let y be a vertex in $N_G(b) \cap V(A)$, whose existence is guaranteed by the assumption that $\{b, A\} \in E(\widehat{H})$. The desired chordless cycle can be defined by the edges $\{v, b\}, \{v, z(A)\}, \{b, y\}$ and the edge-set of a shortest path between $z(A)$ and y in A – in particular, by the definition of \mathcal{A} and Lemma 3.10, v is not adjacent to any vertex on this path, excluding $z(A)$, and since $\{b, A\} \in E(\widehat{H})$, b is not adjacent to $z(A)$.

For the second item, choose $y \in N_G(b) \cap V(A)$ and $y' \in N_G(b) \cap V(A')$, whose existence is guaranteed by the assumption that $\{b, A\}, \{b, A'\} \in E(\widehat{H})$, such that the length of a shortest path between $z(A)$ ($z(A')$) and y (resp. y') in A (resp. A') is minimum. Observe that the cycle defined $\{v, z(A)\}, \{v, z(A')\}, \{y, b\}, \{y', b\}$ and the shortest paths between $z(A)$ and y in A and $z(A')$ and y' in A' , respectively, is a chordless cycle – in particular, by the definition of \mathcal{A} , Lemma 3.10 and since $b, b' \in B_f$, the cycle cannot have a chord containing v . \square

Isolated Vertices in \widehat{H} . We start investigating the bipartite graph \widehat{H} by examining the isolated vertices in \mathcal{A} that it contains. In this context, we need the two following lemmata.

LEMMA 3.13. Let $A \in \mathcal{A}$ be an isolated vertex in \widehat{H} , and denote $z = z(A)$. Then, $N_G(V(A)) = N_G(z) \setminus V(A)$.

PROOF. Since $z \in A$, it is clear that $N_G(z) \setminus V(A) \subseteq N_G(V(A))$. Next, we show that $N_G(V(A)) \subseteq N_G(z) \setminus V(A)$. To this end, consider some vertices $y \in V(A)$ and $u \in N_G(y) \setminus V(A)$, and suppose, by way of contradiction, that $u \notin N_G(z)$. Because A is a connected component of H , it holds that $u \in \{v\} \cup B_v \cup X$. Moreover, because A is isolated in \widehat{H} and yet $u \in N_G(V(A)) \setminus N_G(z)$, it further holds that $u \in X$. However, this results in a contradiction to the statement of Lemma 3.11. \square

LEMMA 3.14. Let $A \in \mathcal{A}$ be an isolated vertex in \widehat{H} , and denote $z = z(A)$. Then, G does not have a chordless cycle that contains the edge $\{v, z\}$.

PROOF. Suppose, by way of contradiction, that G has a chordless cycle C that contains the edge $\{v, z\}$. By Lemma 3.10, z is the only neighbor of v in A , and therefore the cycle C must contain an

edge $\{y, u\}$ such that $y \in A$ and $u \notin A$ for $u \neq v$. By Lemma 3.13, $N_G(V(A)) = N_G(z) \setminus A$. Because $u \in N_G(V(A))$, this means that $\{z, u\} \in E(G)$. Hence, since the cycle C is chordless, it holds that $z = y$, which, in turn, implies that $u \notin N_G(v)$. We deduce that u must belong to B_f . However, this implies that $\{A, u\} \in \widehat{H}$, contradicting the assumption that A is an isolated vertex in \widehat{H} . \square

Lemma 3.14 leads us to the design of the following reduction rule.

REDUCTION RULE 3. *If the graph \widehat{H} contains an isolated vertex $A \in \mathcal{A}$, mark the edge $\{v, z\}$ as irrelevant.*

After an exhaustive application of this rule, we can assume that \widehat{H} does not contain an isolated vertex $A \in \mathcal{A}$.

Applying the Expansion Lemma. Next, we would like to apply Lemma 2.2 in the context of the bipartite graph \widehat{H} . Since $|\mathcal{A}| \geq \Delta - f(k) \geq (k+2) \cdot |B_v|$ and we have already ensured that \widehat{H} does not contain any isolated vertex $A \in \mathcal{A}$, this lemma implies that we can find (in polynomial time) non-empty subsets $\mathcal{A}^* \subseteq \mathcal{A}$ and $B^* \subseteq B_v$ such that there exists a $(k+2)$ -expansion from B^* into \mathcal{A}^* .

The usefulness of B^* is stated in the following lemma.

LEMMA 3.15. *Any solution of size at most k to the input instance that does not contain v contains all of the vertices in B^* .*

PROOF. Suppose that the input instance is a yes-instance, and suppose it has a solution S of size at most k that does not contain v . Consider some vertex $b \in B^*$. Let A_1, A_2, \dots, A_{k+2} be the neighbors of b in \widehat{H} that correspond to our $(k+2)$ -expansion. For any choice of A_i and A_j , Lemma 3.12 implies that if there is no chordless cycle defined by v, b and a path in A_i , then there is a chordless cycle defined by v, b , a path in A_i and a path in A_j . Therefore, if S contains neither v nor b , it must contain at least one vertex from each connected component A_i excluding at most one such component. However, there are $k+2$ such components, and since S does not contain v , we deduce that it contains b . The choice of $b \in B^*$ was arbitrary, and therefore we conclude that S contains all of the vertices in B^* . \square

Decreasing the Independent Degree of v . Armed with Lemma 3.15, we can apply the following reduction rule.

REDUCTION RULE 4. (a) *For each vertex $b \in B^*$, insert the edge $\{b, v\}$ into $E(G)$ (if it is not already present), and mark $\{b, v\}$ as a mandatory edge.*

(b) *Mark each edge $\{v, z(A)\}$ such that $A \in \mathcal{A}^*$ as an irrelevant edge.*

Let us consider the first statement of the reduction rule.

LEMMA 3.16. *Reduction Rule 4(a) is safe.*

PROOF. Suppose that (G, k, E_M, E_I) is the given instance. We claim that it is safe to add mandatory edges between v and all of the vertices in B^* . To this end, let G' denote the graph obtained after applying this operation, and let E'_M denote the new set of mandatory edges. Note that we obtain a new instance (G', k, E'_M, E_I) .

We first prove the forward direction. For this purpose, let S be a solution of size k for the given instance (G, k, E_M, E_I) . By Lemma 3.15, either the vertex v or all of the vertices in B^* must be present in any solution of size at most k in G , and in particular, they must be present in S . Therefore, S includes an endpoint of every new mandatory edge, and hence it hits every mandatory edge in

$E'_M \setminus E_M$. Because $G \setminus S$ and $G' \setminus S$ are equal, we conclude that S hits all chordless cycles in G' and all mandatory edges in E_M . Therefore, S is a solution for (G', k, E'_M, E_I) as well.

In the reverse direction, consider any solution S' of size k to (G', k, E'_M, E_I) . Observe that by the definition of E'_M , S' contains an endpoint of every mandatory edge in E'_M . Now, consider the set S' in the graph G , and note that S' intersects every edge in E_M as $E_M \subseteq E'_M$. Observe that if C is a chordless cycle in G but not in G' , then there must be a newly added mandatory edge that forms a chord of C in G' . Therefore, any chordless cycle C in G either contains a chord in G' that is a mandatory edge, or it is also a chordless cycle in G' . In any case, S' must contain a vertex of C . Therefore, S' is a solution of size k to the instance (G, k, E_M, E_I) . \square

Now, we consider the second statement of the reduction rule.

LEMMA 3.17. *Reduction Rule 4(b) is safe.*

PROOF. Let (G', k, E'_M, E_I) be the instance that is obtained after applying the first statement. Let (G', k, E'_M, E'_I) denote the new instance obtained by marking the edges between v and $\{z(A) \mid A \in \mathcal{A}^*\}$ as irrelevant. Observe that the graph remains unchanged, and therefore any solution S to the instance (G', k, E'_M, E_I) is also a solution to (G', k, E'_M, E'_I) , which proves the forward direction.

In the reverse direction, let S' be a solution to (G', k, E'_M, E'_I) . We argue that if S' is also a solution to (G', k, E'_M, E_I) . Note that S' hits all chordless cycles in G' that do not contain any edge from E'_I , and that it also hits every edge in E'_M . Thus, it remains to show that S' hits all chordless cycles in G' that contain at least one edge from E'_I . To this end, consider a chordless cycle C in G' such that it contains a new irrelevant edge in E'_I . Then, the cycle C must contain v and $z(A)$ for some $A \in \mathcal{A}^*$. If S' contains v , then clearly S' intersects the cycle C . Hence, we next suppose that this is not the case. Then, S' contains all of B^* , since E'_M contains all the edges between v and B^* . We claim that the cycle C contains a vertex from B^* , which would mean that C is hit by S' . Observe that, if the cycle C in the graph G' does not contain any vertex of B^* , then it contains no newly added mandatory edges in E'_M . Hence, C is also a chordless cycle in the graph G that we had at hand before applying part (a) of the reduction rule. This follows from the fact that G and G' differ only in the set of new mandatory edges. Therefore the chordless cycle C , which contains the vertices v and $z(A)$, must also contain a vertex $b \in B_v$, (the v -blocker in G). But then, by the definition of the bipartite graph \widehat{H} , the sets \mathcal{A}^* and B^* , and Lemma 2.2, the vertex b must lie in B^* . Indeed, let $C = v_0, \dots, v_r$ where $v_0 = v, v_1 = z(A)$, and consider it in the graph G . Let $b = v_{i+1}$ be the first vertex from B_v in the sequence v_0, \dots, v_r . Denote $A' = \{v_1, \dots, v_i\}$. Since C is chordless, $v_i \notin N_G(v)$ for $i = 2, \dots, l$. Hence, the edges of the path v_1, \dots, v_i are in H and $A' \subseteq A$. Since $v_i b \in E$, we have that $Ab \in E(\widehat{H})$, and by the second condition in Lemma 2.2, we conclude that $b \in B^*$. We thus get that C is hit by S' . Hence, S' is a solution to (G', k, E'_M, E_I) . \square

Reduction Rule 4 decreases $|N_G^R(v)|$. Moreover, as long as $d_G^I(v) > \Delta$, we can apply this rule. Thus, after an exhaustive application of this rule, it should hold that $d_G^I(v) \leq \Delta$. Furthermore, this rule neither inserts vertices into $V(G)$ nor unmarks edges, and therefore we conclude that Lemma 3.7 is correct. Denote $\Delta' = \Delta + k$. Thus, by Reduction Rule 2, the size of a maximum independent set in the neighborhood of each vertex in the graph G from which we remove irrelevant edges is bounded by Δ' .

3.4 The Clique Forest

In the rest of the paper, we focus on reducing the graph using the properties of clique-forests. Let F denote the clique forest associated with the chordal graph $G \setminus D'$. In the following sections, for a vertex $v \in G \setminus D'$, we often define B_v as a certain bag in the clique-forest F that contains v . Recall

that B_v was earlier defined to denote a v -blocker in G . However, this abuse of notation introduces no ambiguity as v -blockers are not required in the following sections, and Furthermore it improves the readability. Let us now proceed towards bounding the number of leaves in F . We start the following lemma.

LEMMA 3.18. *Let I be an independent set in the graph $G \setminus D'$. Then, there are most $|D'| \cdot \Delta'$ relevant edges between vertices in D' and vertices in I .*

PROOF. The claim follows from the observation that since $\Delta_G^I \leq \Delta$, $|E_M| \leq k^2$ (as Reduction rule 2 doesn't apply) and I is an independent set, for every vertex $v \in D'$, there are at most Δ' relevant edges between v and vertices in I . \square

We will also need the following reduction rule.

REDUCTION RULE 5. *If there exists a vertex v in $G \setminus D'$ such that the vertices in $N_G(v)$ which are connected to v via relevant edges form a clique, remove the vertex v from G .*

LEMMA 3.19. *Reduction Rule 5 is safe.*

PROOF. The special choice of the vertex v implies that every chordless cycle containing v must contain at least one irrelevant edge, and can therefore be ignored. We thus conclude that it is safe to remove the vertex v from G . \square

The bound on the number of leaves will follow from a bound on the number of bags containing private vertices, which are defined as follows.

Definition 3.20. A vertex v in $G \setminus D'$ is a private vertex if there exists only one bag in F that contains it.

LEMMA 3.21. *The number of bags in F containing private vertices is bounded by $|D'| \cdot \Delta'$.*

PROOF. Let ℓ denote the number of bags in F containing private vertices. By the definition of a clique forest, if we take exactly one private vertex from each bag in F , we obtain an independent set. Therefore, the graph G contains an independent set of size at least ℓ . Let I denote this independent set. Observe that since each vertex in I is a private vertex, its neighborhood in $G \setminus D'$ forms a clique. Therefore, after an exhaustive application of Reduction Rule 5, each vertex in I must be connected by at least one relevant edge to a vertex in D' . By Lemma 3.18, we conclude that $\ell \leq |D'| \cdot \Delta'$. \square

We are now ready to bound the number of leaves and nodes of degree at least 3 in the clique forest F .

LEMMA 3.22. *Both the number of leaves in F and the number of nodes of degree at least 3 in F are bounded by $|D'| \cdot \Delta'$.*

PROOF. Observe that since every leaf in F corresponds to a maximal clique in $G \setminus D'$, it contains a private vertex. Thus, by Lemma 3.21, the number of leaves is bounded by $|D'| \cdot \Delta'$. Since in a forest, the number of nodes of degree at least 3 is bounded by the number of leaves, we conclude that the lemma is correct. \square

Next, we turn to bound the size of a bag of F , to which end we prove the correctness of the following lemma.

LEMMA 3.23. *In polynomial time we can produce an instance (G', k', E'_M, E'_I) equivalent to (G, k, E_M, E_I) such that $k' \leq k$ and the size of any maximal clique in G' is bounded by $\kappa = c \cdot (|\bar{D}|^3 \cdot k + |\bar{D}| \cdot \Delta' \cdot (k+2)^3)$. (Here c is some constant independent of the input.)*

The proof of this lemma closely follows the approach using which [36] bounds the size of maximal cliques, but requires a few modification since the purpose of the paper [36] is to give a parameterized algorithm for CVD, which runs in exponential time, while we need to reduce instances in polynomial time. The details are deferred to Section 4, since it is an adaption of the work of Marx and is largely independent of the results of this section.

From now onwards, let us consider (G, k, E_M, E_I) as the instance obtained after the application of the above lemma. The redundant solution D' for this instance is defined as before. Observe that the size of each bag of F is bounded by the size of a maximal clique of $G \setminus D'$. Furthermore, since $G \setminus D'$ is a subgraph of G , the size of a maximal clique of $G \setminus D'$ is bounded by the size of a maximal clique of G . Thus, having applied the procedure given by Lemma 3.23, we have the following result.

LEMMA 3.24. *The size of any bag of F is upper bounded by κ .*

3.5 The Length of Degree-2 Paths

The Family of Paths \mathcal{P} . Let V_F denote the set consisting of each node of degree at least 3 in the forest F as well as each node whose bag has at least one private vertex. Let \mathcal{P} denote the set of paths in the forest F whose endpoints belong to V_F and such that all of their internal nodes do not belong to V_F . Clearly, it holds that $|\mathcal{P}| \leq |V_F|$. By Lemmata 3.21 and 3.22, we have the following observation.

OBSERVATION 1. $|\mathcal{P}| \leq 2|D'| \cdot \Delta'$.

Thus, in light of Lemma 3.24, by bounding the maximum number of nodes on each path in \mathcal{P} , we can bound the total number of vertices in the graph. To this end, we fix some path $P \in \mathcal{P}$. Moreover, we orient the path from left to right, where the choice of the leftmost and rightmost nodes is arbitrary.

Partitioning the Path P . Next, we will partition P into more “manageable paths”. To this end, we need the following definition.

Definition 3.25. We say that a subpath Q of P complies with a vertex $d \in D'$ if at least one of the following conditions holds.

- (1) For every two bags B and B' on Q , both $B \subseteq N_G(d)$ and $B' \subseteq N_G(d)$.
- (2) Let B_1, B_2, \dots, B_t denote the bags of Q ordered from left to right. Then, at least one of the following condition holds.
 - (a) $B_1 \cap N_G(d) \subseteq B_2 \cap N_G(d) \subseteq \dots \subseteq B_t \cap N_G(d)$.
 - (b) $B_t \cap N_G(d) \subseteq B_{t-1} \cap N_G(d) \subseteq \dots \subseteq B_1 \cap N_G(d)$.

In particular, $N_G(d) \cap (\cup_{i=1}^t B_i)$ is a subset of at least one of the two bags B_1 and B_t .

We would like to find a set \mathcal{B} of at most $\mathcal{O}(|D'|)$ bags on the path P such that after their removal from P , the following lemma will be true.

LEMMA 3.26. *Each subpath resulting from the removal of the bags in \mathcal{B} from P complies with every vertex in D' .*

The rest of this subsection concerns the proof of this lemma. To prove it, it is sufficient to show that for each vertex $d \in D'$, we can find $\mathcal{O}(1)$ bags such that after their removal from P , each of the resulting subpaths complies with d . To this end, fix some vertex $d \in D'$.

First, we need the following lemma.

LEMMA 3.27. *Let $u, v \in N_G(d) \setminus D'$ be non-adjacent vertices, B_u be a bag containing u such that no bag to its right (on P) contains u , B_v be a bag containing v such that no bag to its left (on P) contains*

v , and B_u lies on the left of B_v (on P). Then, d is adjacent to every vertex in every bag that lies strictly between B_u and B_v .

PROOF. Let z be a vertex in a bag that lies strictly between B_u and B_v . Since this bag is not an endpoint of the path P , z belongs to at least two bags that lie between B_u and B_v (here we rely on the construction of V_F and \mathcal{P}). Denote the path between B_u and B_v by $B_u = B_1 - B_2 - B_3 - \dots - B_t = B_v$. Let B_i be the leftmost bag containing z , and let B_j be the rightmost bag containing z . Suppose, by way of contradiction, that $z \notin N_G(d)$.

We claim that there exists a path in G , P_1 , from u to z such that none of its internal vertices belongs to $B_{i+1} \cup B_{i+2} \dots \cup B_v$. The proof by induction on the number i . When $i = 1$, we have that $B_i = B_u$. Then, since the vertices in B_u form a clique, the claim is correct. Now, suppose that the claim holds for $i - 1 \geq 1$, and let us prove it for i .

Since the vertices of the bag B_i form a maximal clique in the graph $G \setminus D'$, and the bag B_i does not contain private vertices, we have that $B_i \subseteq B_{i-1} \cup B_{i+1}$ and $B_i \setminus B_{i+1} \subset B_{i-1}$ is non-empty. Consider a vertex $w \in B_i \setminus B_{i+1}$ and observe that z and w are adjacent in G (since both belong to B_i), and further the vertex w is not present in $B_{i+1} \cup B_{i+2} \dots \cup B_v$. By induction, there is a path P from u to w whose internal vertices do not belong to $B_i \cup B_{i+1} \dots \cup B_v$. Appending the edge (w, z) to P gives us the path P_1 .

Similarly, there is a path P_2 from z to v such that none of its internal vertices belongs to $B_u \cup \dots \cup B_{j-2} \cup B_{j-1}$. Since $i < j$, the paths P_1 and P_2 have no common vertex except z , and there is no edge between $V(P_1) \setminus \{z\}$ and $V(P_2) \setminus \{z\}$.

Let P'_1 be the subpath of P_1 from u' to z , where u' is the last vertex in P_1 adjacent to d . Similarly, let P'_2 be the subpath of P_2 from z to v' , where v' is the first vertex in P_2 adjacent to d . We may assume that P'_1 and P'_2 do not contain chords, else we can replace P'_1 and P'_2 by a chordless subpath of P'_1 and chordless subpath of P'_2 , respectively, which will still contain u' , z and v' . The cycle $d - P'_1 - z - P'_2 - d$ is a chordless cycle in G , contradicting the fact that $G \setminus (D' \setminus \{d\})$ is a chordal graph. \square

We also need the following notation. Let B_ℓ be the leftmost bag on P that contains a neighbor v_ℓ of d such that v_ℓ does not belong to any bag to the right of B_ℓ . Similarly, let B_r be the rightmost bag on P that contains a neighbor v_r of d such that v_r does not belong to any bag to the left of B_r .

LEMMA 3.28. *Let B and B' be two bags on P that do not lie on the right of B_ℓ and such that B lies on the left of B' . Then, it holds that $B \cap N_G(d) \subseteq B' \cap N_G(d)$.*

PROOF. Suppose, by way of contradiction, that $B \cap N_G(d) \not\subseteq B' \cap N_G(d)$. However, this implies that B contains a neighbor v of d such that v does not belong to any bag to the right of B , which contradicts the choice of B_ℓ . \square

LEMMA 3.29. *Let B and B' be two bags on P that do not lie on the left of B_r and such that B lies to the right of B' . Then, it holds that $B \cap N_G(d) \subseteq B' \cap N_G(d)$.*

PROOF. The proof of this lemma is symmetric to the proof of Lemma 3.28. \square

By Lemmas 3.27–3.29, each of the subpaths resulting from the removal of B_ℓ and B_r from P complies with d . Thus, we conclude that Lemma 3.26 is correct.

Handling a Manageable Path. We now examine a subpath of P , denoted by Q , which complies with every vertex $d \in D'$. We will devise reduction rules such that after applying them exhaustively, the number of vertices in the union of the bags of the path Q will be bounded by $\mathcal{O}(\kappa)$.

Let B_1, B_2, \dots, B_t denote the bags of Q ordered from left to right. Moreover, denote $V(Q) = \bigcup_{i=1}^t B_i$ and $A = \bigcap_{i=1}^t B_i$. We partition D' into two sets D_a and D_p , where $D_a = \{d \in D' : V(Q) \subseteq$

$N_G(d)$ and $D_p = D' \setminus D_a$. Here the letters a and p stand for “all” and “partial”, respectively. Definition 3.25 directly implies that the following observation is correct.

OBSERVATION 2. *For every vertex $d \in D_p$, either (i) $B_1 \cap N_G(d) \subseteq B_2 \cap N_G(d) \subseteq \dots \subseteq B_t \cap N_G(d)$ or (ii) $B_t \cap N_G(d) \subseteq B_{t-1} \cap N_G(d) \subseteq \dots \subseteq B_1 \cap N_G(d)$. In particular, either (i) $N_G(d) \cap V(Q) \subseteq B_1$ or (ii) $N_G(d) \cap V(Q) \subseteq B_t$.*

We also denote $U = V(Q) \setminus (B_1 \cup B_t)$. It is sufficient to ensure that $|U| = \mathcal{O}(\kappa)$ since then, by Lemma 3.24, $|V(Q)| = \mathcal{O}(\kappa)$. Thus, we can next suppose that $|U| > \delta$ where $\delta = 2(k+1) + 6\kappa$.

For each pair of non-adjacent vertices in D_a , we apply Reduction Rule 1. The correctness of this operation is given by the following lemma.

LEMMA 3.30. *Let u and v be two distinct non-adjacent vertices in D_a . Then, every solution of size at most k contains at least one of the vertices u and v .*

PROOF. Since $|U| > \delta$ and the size of each bag is bounded by κ , standard arguments on weighted paths imply that there exists $i \in [t]$ such that $|\bigcup_{j=1}^i B_j| > \delta/2 - \kappa$ and $|\bigcup_{j=i+1}^t B_j| > \delta/2 - \kappa$; indeed, we may iteratively increase i from 1 to t until we reach the first time where it holds that $|\bigcup_{j=1}^i B_j| > \delta/2 - \kappa$, in which case it will also hold that $|\bigcup_{j=i+1}^t B_j| > \delta/2 - \kappa$. Denote $U^1 = U \cap ((\bigcup_{j=1}^{i-1} B_j) \setminus (B_1 \cup B_i))$ and $U^2 = U \cap ((\bigcup_{j=i+2}^t B_j) \setminus (B_{i+1} \cup B_t))$. Then, again since the size of each bag is bounded by κ , it holds that $|U^1|, |U^2| > \delta/2 - 3\kappa \geq k+1$. Moreover, by the definition of a clique forest, $U^1 \cap U^2 = \emptyset$ and there is no vertex in U^1 that is adjacent to a vertex in U^2 . Thus, for any pair of vertices $x \in U^1$ and $y \in U^2$, the subgraph induced by $\{u, v, x, y\}$ is a chordless cycle. However, any solution of size at most k can only contain at most k vertices from $U^1 \cup U^2$, and thus, to hit all of these chordless cycles, it must contain at least one vertex among u and v . \square

Thus, from now on we can assume that $G[D_a]$ is a clique. However, by the definition of A , for every vertex in A and every vertex in $V(Q)$, there exists a bag B_i , $i \in [t]$, which contains both of them, and therefore they are adjacent. We thus deduce that the following observation is correct.

OBSERVATION 3. *Any two distinct vertices $v \in D_a \cup A$ and $u \in D_a \cup V(Q)$ are adjacent.*

Let us now examine chordless cycles that contain vertices from U .

LEMMA 3.31. *Let C be a chordless cycle in G that contains some vertex $u \in U$. Then, no vertex on $V(C)$ belongs to $D_a \cup A$, and both neighbors of u in C do not belong to $D' \cup A$.*

PROOF. First, by Observation 2, we have that u does not have neighbors (in G) in D_p , and therefore both the neighbors of u in C do not belong to D_p . Thus, it remains to show that no vertex in $V(C)$ belongs to $D_a \cup A$. By Observation 3, if at least one of vertices $v \in V(C)$ belongs to $D_a \cup A$, it is adjacent to u in G and therefore it is a neighbor of u in C . Moreover, the other neighbor w of u in C is not in D_a for otherwise vw is a chord by Observation 3. Hence $w \in V \setminus D'$ and since $uw \in E$, $w \in \bigcup_{i=2}^{t-1} B_i$ and in particular $w \in V(Q)$. It follows that by Observation 3 vw is a chord, a contradiction. \square

LEMMA 3.32. *Let C be a chordless cycle in G that contains some vertex $u \in U$. Then, C contains a path between a vertex in $B_1 \setminus A$ and a vertex in $B_t \setminus A$ whose internal vertices belong to U and one of them is u .*

PROOF. Since D' is an approximate solution, the cycle C must contain at least one vertex that does not belong to U . And by Lemma 3.31, C is disjoint from $A \cup D_a$. Hence the cycle C contains a vertex in D_p . This implies that C contains two subpaths, each between u and a vertex in $(B_1 \cup B_t) \setminus A$, whose only common vertex is u and whose internal vertices belong to U . Moreover, one of these

paths must contain an endpoint from $B_1 \setminus A$ and the other from $B_t \setminus A$, else the cycle C contains a chord corresponding to the edge between these two endpoints. Therefore, by concatenating the two paths, we obtain the desired subpath of C . \square

We continue our examination of chordless cycles that contain vertices from U in the context of separators.

LEMMA 3.33. *Let S be a minimal solution that contains at least one vertex from U . Then, there exists $i \in [t - 1]$ such that (i) $(B_i \cap B_{i+1}) \setminus A \subseteq S$, and (ii) $S \cap U \subseteq B_i \cap B_{i+1}$.*

PROOF. **Property (i).** Since S is a minimal solution, G contains a chordless cycle C with a vertex $u \in S \cap U$ and no other vertex from S . By Lemma 3.32, C contains a path P between a vertex in $x \in B_1 \setminus A$ and a vertex in $y \in B_t \setminus A$ whose set of internal vertices includes u and is a subset of U . In particular, since each bag induces a clique while C is a chordless cycle, the internal vertices of the path P is a non-empty subset of $V(Q) \setminus A$. Moreover $(V(C) \setminus V(P)) \cap V(Q) = \emptyset$, since the path P contains a vertex from each bag B_i for $i = 1, 2, \dots, t$. This also implies that x and y are not adjacent. Moreover, by Lemma 3.31, $V(C) \cap (D_a \cup A) = \emptyset$. Thus, since D' is a redundant approximate solution, C contains a vertex $d \in D_p$.

Suppose, by way of contradiction, that there does not exist $i \in [t - 1]$ such that $(B_i \cap B_{i+1}) \setminus A \subseteq S$. Then, G has a (chordless) path P' between x and y whose internal vertices belong to $V(Q) \setminus (S \cup A)$. Since $(V(C) \setminus V(P)) \cap V(Q) = \emptyset$, it holds that $(V(C) \setminus V(P)) \cap V(P') = \emptyset$, which implies that C contains a path between x and y whose set of internal vertices is disjoint from the one of $V(P')$. Observe that if a graph H contains a vertex a with two non-adjacent neighbors, b and c , such that $H \setminus \{a\}$ has a chordless path between b and c with at least one vertex that is not a neighbor of a , then H has a chordless cycle. On the one hand, by the definition of a clique forest, any vertex in $V(G) \setminus (V(Q) \cup D')$ cannot be adjacent to both a vertex in $B_1 \setminus A$ and a vertex in $B_t \setminus A$, and further it is adjacent to no vertex in U . On the other hand, Observation 2 implies that any vertex in D_p also satisfies this property. Thus, $V(P') \cap U = \emptyset$, since any vertex in this set fits the above description of the vertex a where $H = G[(V(C) \setminus V(P)) \cup V(P')]$, which is a subgraph of the chordal graph $G \setminus S$. Without loss of generality, we have that P' contains a subpath $p - q - r$ where $p, q \in B_1 \setminus A$ and $r \in B_t \setminus A$. Let P'' denote a shortest path between p and r in $G[(V(C) \setminus V(P)) \cup (V(P') \setminus \{q\})]$, which contains at least two internal vertices (since it must contain a vertex from $V(C) \setminus V(P)$, else P' would have had a chord, and such a vertex cannot be adjacent to both p and r). Every vertex on P'' should be adjacent to q , else q fits the above description of the vertex a . Since P' is a chordless path and $(V(C) \setminus V(P)) \cap V(Q) = \emptyset$, P'' does not contain any vertex from B_1 that is not p , and the only vertex from B_t that is not r and which P'' can contain is the neighbor of r . Overall, this implies that P'' contains a vertex from $V(C) \setminus V(P)$ adjacent to both q and a vertex in $B_t \setminus A$, which is a contradiction.

Property (ii). Pick any $i \in [t - 1]$ such that $(B_i \cap B_{i+1}) \setminus A \subseteq S$. Now we will show that $S \cap U \subseteq B_i \cap B_{i+1}$. To this end, we consider some arbitrary vertex $u \in S \cap U$ and show that it belongs to $B_i \cap B_{i+1}$. Since S is a minimal solution, G contains a chordless cycle C' such that $V(C') \cap S = \{u\}$. By Lemma 3.32, C' contains a subpath between a vertex in $B_1 \setminus A$ and a vertex in $B_t \setminus A$ whose internal vertices belong to U . Thus, by the definition of a clique forest, C' must contain a vertex from $(B_i \cap B_{i+1}) \setminus A$. Since $(B_i \cap B_{i+1}) \setminus A \subseteq S$, and therefore it must hold that $u \in B_i \cap B_{i+1}$ (since otherwise we reach a contradiction to the fact that $V(C') \cap S = \{u\}$). \square

Let \mathcal{W} be the family of each subset $W \subseteq (B_1 \cup B_t) \setminus A$ of size at most k for which there exists an index $i \in [t - 1]$ such that $W = (B_i \cap B_{i+1}) \setminus (A \cup U)$ and $|(B_i \cap B_{i+1}) \cap U| \leq k$. We can easily bound the size of the family \mathcal{W} as follows.

LEMMA 3.34. $|\mathcal{W}| \leq 2k + 1$.

PROOF. Let i be the smallest index in $[t - 1]$ for which there exists $W^i \in \mathcal{W}$ such that $W^i = (B_i \cap B_{i+1}) \setminus (A \cup U)$, and let j be the largest for which there exists $W^j \in \mathcal{W}$ such that $W^j = (B_j \cap B_{j+1}) \setminus (A \cup U)$. Then, by the definition of a clique forest, for every set $W \in \mathcal{W}$ it holds that $W \subseteq W^i \cup W^j$. Furthermore, the sets in \mathcal{W} can be sorted by $W_1, W_2, \dots, W_{|\mathcal{W}|}$ such that for all $r \in [|\mathcal{W}| - 1]$, $W_{r+1} \cap B_1 \subseteq W_r \cap B_1$ and $W_r \cap B_t \subseteq W_{r+1} \cap B_t$. We thus conclude that $|\mathcal{W}| \leq 2k + 1$. \square

We proceed by associating a separator with each set $W \in \mathcal{W}$ as follows. First, let \mathcal{I}_W denote the set of all indices $i \in [t - 1]$ such that $W = (B_i \cap B_{i+1}) \setminus (A \cup U)$. Now, let i_W denote an index in \mathcal{I}_W that minimizes $|(B_i \cap B_{i+1}) \cap U|$ (if there are several choices, choose one arbitrarily). We further denote $M = \bigcup_{W \in \mathcal{W}} ((B_{i_W} \cap B_{i_W+1}) \cap U)$. Observe that by Lemma 3.34 and the definition of \mathcal{W} , $|M| = \mathcal{O}(k^2)$. Thus, it is sufficient to argue that there exists a vertex in $U \setminus M$ that can be removed from G (since as long as $|U| > \delta$, we will be able to find such a vertex). To this end, we will need the following lemma.

LEMMA 3.35. *Let $u \in U \setminus M$. If (G, k) is a yes-instance, then it has a solution S of size at most k that does not contain the vertex u .*

PROOF. Suppose that (G, k) is a yes-instance, and let S be a solution of minimum size. Assume that $u \in S$, else we are done. By Lemma 3.33, there exists $i \in [t - 1]$ such that $(B_i \cap B_{i+1}) \setminus A \subseteq S$ and $S \cap U \subseteq B_i \cap B_{i+1}$. Denote $W = (B_i \cap B_{i+1}) \setminus (A \cup U)$. Since $|S| \leq k$ and $W \subseteq S$, we have that $W \in \mathcal{W}$. Denote $R = (B_{i_W} \cap B_{i_W+1}) \cap U$ and $T = (B_i \cap B_{i+1}) \cap U$. Since $u \in U \setminus M$, it holds that $u \notin R$. Moreover, since $S \cap U \subseteq B_i \cap B_{i+1}$, it holds that $u \in T$. By the definition of i_W , we have that $|R| \leq |T|$. Thus, to show that the lemma is correct, it is sufficient to show that $S' = (S \setminus T) \cup R$ is a solution.

Suppose, by way of contradiction, that S' is not a solution. Then, since S is a solution of minimum size, there exist a chordless cycle C and a vertex $v \in T$ such that $v \in V(C)$ and $V(C) \cap S' = \emptyset$. Since $T \subseteq U$, by Lemma 3.32, C contains a path between a vertex in $(B_1 \cap B_2) \setminus A$ and a vertex in $(B_{t-1} \cap B_t) \setminus A$ whose internal vertices belong to U . In particular, by the definition of a clique forest, $V(C) \cap ((B_{i_W} \cap B_{i_W+1}) \setminus A) \neq \emptyset$. However, we have that $(B_{i_W} \cap B_{i_W+1}) \setminus A = R \cup W \subseteq S'$, which contradicts the fact that $V(C) \cap S' = \emptyset$. \square

We are now ready to present our reduction rule.

REDUCTION RULE 6. *Let $u \in U \setminus M$. Remove the vertex u from the graph G and add an edge between any two non-adjacent vertices in $N_G(u)$.*

LEMMA 3.36. *Reduction Rule 6 is safe.*

PROOF. Let G' be the graph resulting from the application of this rule. For the forward direction, suppose that (G, k) is a yes-instance, and let S be a solution of minimum size. By Lemma 3.35, we can assume that $u \notin S$, and therefore $S \subseteq V(G')$. Thus, to show that (G', k) is a yes-instance, we need to prove that S hits every chordless cycle in G' . Let C be a chordless cycle in G' . Suppose that this cycle does not exist in G , else it is clear that S hits it. Then, C contains an edge between two vertices $v, w \in N_G(u)$ that are non-adjacent in G . Observe that since C is a chordless cycle and $G'[N_G(u)]$ is a clique, $V(C) \cap N_G(u) = \{v, w\}$. Thus, by replacing $\{v, w\}$ by $\{v, u\}$ and $\{u, w\}$, we obtain a chordless cycle in G . Since S hits this cycle and $u \notin S$, it holds that S also hits C .

For the backward direction, suppose that (G', k) is a yes-instance, and let S be a solution of minimum size. To show that (G, k) is a yes-instance, it is sufficient to show that S is also a solution to (G, k) . Since $V(G') \subseteq V(G)$, it is clear that $S \subseteq V(G)$. We need to prove that S hits every chordless

cycle in G . Let C be a chordless cycle in G . Suppose that this cycle does not exist in G' , else it is clear that S hits it. Furthermore, suppose that $G'[V(C) \setminus \{u\}]$ does not contain a chordless cycle, else again it is clear that S hits C . We get that C contains two vertices $v, w \in N_G(u)$ that are not adjacent in G . Since $u \in U$, Observation 2 and the definition of a clique forest imply that $N_G(u) \subseteq V(Q) \cup D_a$. By Observation 3, we deduce that $v, w \notin D_a \cup A$, and that C contains at most one vertex from $D_a \cup A$ (since any two vertices from $D_a \cup A$ are adjacent to each other and to both v and w). Since D' is a redundant approximate solution, C must contain a vertex $p \in D_p$. Since $G'[V(C) \setminus \{u\}]$ does not contain a chordless cycle, the neighbors of p on C belong to $N_G(u)$, and we can assume w.l.o.g. that these neighbors are v and w . However, since v and w are not adjacent there cannot be a bag that contains both of them, which results in a contradiction to Observation 2. \square

3.6 Unmarking Irrelevant and Mandatory Edges

Recall that our instance includes a set E_I of irrelevant edges and a set E_M of mandatory edges. It is clear that we can unmark each irrelevant edge (these edges were marked only for the sake of clarity of the analysis of our kernel). However, to unmark mandatory edge, we need the following operation.

REDUCTION RULE 7. *For every mandatory edge $\{x, y\}$ introduce $k + 1$ pairs of new vertices, $\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_{k+1}, y_{k+1}\}$, and for each pair $\{x_i, y_i\}$ add the edges $\{x, x_i\}, \{x_i, y_i\}$ and $\{y_i, y\}$. Moreover, unmark the edge $\{x, y\}$.*

LEMMA 3.37. *Reduction Rule 7 is safe.*

PROOF. Let $(G', k, E_I = \emptyset, E'_M)$ be the instance resulting from the application of this rule. Each edge $\{x, y\}$ is contained (in G') in $k + 1$ cycles of size 4 which do not share vertices other than x and y . Therefore, and solution of size at most k to (G', k) must contain at least one of the vertices x and y . Thus, since any chordless cycle in G is also present in G' and we have only unmarked the edge $\{x, y\}$, we conclude that if (G', k, E_I, E'_M) is a yes-instance, so is (G, k, E_I, E_M) . On the other hand, any solution to (G, k, E_I, E_M) hits all of the chordless cycles in G' since each of these chordless cycles is either present in G or contain both of the vertices x and y . Therefore, if (G, k, E_I, E_M) is a yes-instance, so is (G', k, E_I, E'_M) . \square

Recall that $|E_M| \leq k^2$. Hence, the total number of newly added vertices does not exceed $O(k^3)$.

3.7 The Number of Vertices in the Kernel

In this section, we obtained an approximate solution \tilde{D} of size $f(k)$ and a redundant approximate solution D of size $O(k \cdot f(k))$. Then, we examined the clique forest F associated with the chordal graph $G \setminus D'$ where $|D'| = O(|D| + k^2)$. To this end, we considered a set \mathcal{P} of degree-2 paths that together cover all of the nodes of the forest, and showed that $|\mathcal{P}| = O(|D'| \cdot \Delta')$. Recall that $\Delta' = O(k \cdot f(k))$. We removed $O(|D'|)$ bags, each of size $\kappa = O(|\tilde{D}|^3 \cdot k + |\tilde{D}| \cdot \Delta' \cdot k^3)$, from each path $P \in \mathcal{P}$, and considered each of the resulting subpaths Q . We showed the number of vertices in the union of the bags of the path Q will be bounded by $O(\kappa)$. Finally, we added $O(k^3)$ new vertices to unmark mandatory edges. Thus, we conclude that the number of vertices in our kernel is bounded by

$$\begin{aligned}
& O(|\mathcal{P}| \cdot |D'| \cdot \kappa + |D'| \cdot \Delta' \cdot \kappa + k) \\
= & O(|\mathcal{P}| \cdot |D'| \cdot \kappa) \\
= & O(|D'|^2 \cdot \Delta' \cdot (|\tilde{D}|^3 \cdot k + |\tilde{D}| \cdot \Delta' \cdot k^3)) \\
= & O(f(k)^3 k^3 \cdot (f(k)^3 k + f(k)^2 k^4)) \\
= & O(f(k)^5 k^4 \cdot (f(k) + k^3))
\end{aligned}$$

Recall that by Lemma 3.1, we can assume that $f(k) = O(k^3 \log^2 k)$. Thus, at this point, we obtain a kernel of size $O(k^{22} \log^{12} k)$.

3.8 A Better Kernelization Algorithm

Finally, we present a bootstrapping trick that will exploit the nature of our approximation algorithm to obtain a kernel of size $O(k^{12} \log^{10} k)$. Recall that at this point, where we have already run our kernelization algorithm once, it holds that $n = O(k^{22} \log^{12} k)$. Now, we again recall that CVD admits an $O(\log^2 n)$ -factor approximation algorithm [3]. Currently, it holds that $f(k) = O(k \log^2 k)$ rather than $f(k) = O(k^3 \log^2 k)$. Thus, if we rerun our kernelization procedure, obtaining a kernel of size $O(f(k)^5 k^4 \cdot (f(k) + k^3))$ (see Section 3.7), it now holds that this size is upper bounded by $O(k^{12} \log^{10} k)$. This concludes the proof of correctness of Theorem 1.1.

4 BOUNDING THE SIZE OF MAXIMAL CLIQUES

Let (G, k, E_M, E_I) be an instance of CVD whose independent degree is bounded Δ , and let D be some (possibly approximate) solution to this instance. Here E_M and E_I denote the set of mandatory and irrelevant edges, respectively, of the instance. In this section, we will bound the size of a maximal clique in the graph G . We will show that if the graph admits a chordal deletion set of size at most k , then for any maximal clique K in $G \setminus D$, it is safe to remove all but a bounded number vertices of K . Formally, we will show the following lemma, which is a restatement of Lemma 3.23 along with all the relevant details.

LEMMA 4.1. *Let (G, k, E_M, E_I) be an instance of CVD whose independent degree is bounded Δ , and let D be a solution to this instance. Then in polynomial time we can produce an equivalent instance (G', k', E'_M, E'_I) such that $k' \leq k$, and the size of any maximal clique in G' is bounded by $c \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k + 2)^3)$. (Here c is some constant independent of the input.)*

Our proof of this lemma is an adaptation of the work of [36] with a few modifications. Specifically, the lemmas in [36] construct a so called “necessary set” of vertices with the property that one of the vertices in this set must be part of any solution of size k . We modify these lemmas to ensure that a necessary set output by them always has at most two vertices. Observe that such a necessary set is either a vertex that must be part of any solution of size at most k , or a mandatory edge. We note that [26] also give similar result, inspired by the results of [36]. Given a redundant approximate solution \widehat{D} , the size of a maximal clique in $G \setminus \widehat{D}$ can be bounded by $O(|\widehat{D}|^3 k)$. However the present method for computing a redundant approximate solution implies that $|\widehat{D}| \geq |D| \cdot k$, and hence Lemma 4.1 gives a better upper-bound.

LEMMA 4.2 ([26]). *The size of each bag in the clique forest F is bounded by $c \cdot |\widehat{D}|^3 k$, where c is some constant independent of the input.*

For the rest of this subsection, we fix a maximal clique K of $G \setminus D$, which contains more than $c \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k + 2)^3)$ vertices. We will show that we can mark a bounded number of vertices of K so that the following holds. Let X be any set of at most k vertices such that $G \setminus X$ has a chordless cycle H that contains a vertex u of K . If u is an unmarked vertex then there is another chordless cycle H' in $G \setminus X$ that avoids u and contains strictly fewer unmarked vertex in K . This condition implies that we can safely ignore any chordless cycle that includes an unmarked vertex, which further implies that it is safe to delete these vertices from the graph. We shall closely follow the notations and proofs of [36], but in light of the bound on (relevant) independent-degree of vertices in D , we shall modify them appropriately. For a vertex $v \in V(G) \setminus D$, we say that the vertex has the label $t \in D$ if v is a neighbor of t . Note that the edge (t, v) could be relevant or irrelevant, and a vertex may have several labels depending on its set of neighbors in D .

4.1 Dangerous vertices and their witnesses in K

Let us begin with the notion of dangerous vertices for the clique K in the graph.

Definition 4.3. Let $v \in V(G) \setminus (D \cup K)$ be a relevant neighbor of $t \in D$ such that there is a path P from v to $u \in K$ whose internal vertices do not have the label t .

- (i) The vertex v is a t -dangerous vertex for K if the vertex u does not have the label t .
- (ii) The vertex v is a t^* -dangerous vertex for K if the vertex u is also a relevant neighbor of t and u, v are not neighbors in G .

The vertex $u \in K$ is called a t -witness (t^* -witness) of v , and the path P is called a t -witness (t^* -witness) path of v .

In [36], it is shown that there are many vertices in the clique K whose deletion does not affect any of the dangerous vertices in the following sense. We mark a bounded number of vertices in K such that, for any subset X of vertices of size at most k , if there is chordless cycle in $G \setminus X$ that passes through t , a dangerous vertex v and through some unmarked vertex u in K , then there is another chordless cycle in $G \setminus X$ which contains a marked vertex u' and avoids u . This implies that we may ignore any chordless cycles in G that includes an unmarked vertex of K . Note that the definition above differs from [36] in the requirement that v (and u) must be a relevant neighbor of t , since if the edge (t, v) (or the edge (t, u)) were irrelevant then any chordless cycles that contain both t and v (or t and u) can be safely ignored. We have the following bounds on the size of an independent set of dangerous vertices in G .

LEMMA 4.4 (LEMMA 11, [36]). *If I is any collection of independent t -dangerous vertices, then either $|I| \leq 6k^2$ or we can find a new mandatory edge in the graph, or we can find a vertex that must be part of any solution of size at most k .*

The following lemma improves upon Lemma 12 of [36] by using the bound on the independent degree of vertices in D .

LEMMA 4.5 (LEMMA 12, [36]). *If I is any collection of independent t^* -dangerous vertices, then $|I| \leq \Delta$.*

The following lemma shows that if Q is a clique of t -dangerous vertices we require only $k + 1$ vertices as witnesses for all the vertices of Q .

LEMMA 4.6 (LEMMA 13 [36]). *Let Q be a clique of t -dangerous vertices. Then we can mark $k + 1$ vertices in K such that for any set X of k vertices, if $v \in Q$ has an unmarked t -witness in $K \setminus X$ then it has a marked t -witness in $K \setminus X$.*

In the context of the following lemma, we remark that there is a similar lemma for cliques of t^* -dangerous vertices in [36], but we give a version of it that is more suitable for our purposes.

LEMMA 4.7 (LEMMA 14, [36]). *Let Q be a clique of t^* dangerous vertices. Then either we can find a vertex that must be part of any solution of size at most k , or we can find a new mandatory edge, or else we can mark $(k + 2)^3$ vertices of K such that for any set X of k vertices, if $v \in Q$ has a unmarked t^* witness in $K \setminus X$ then it also has a marked witness $K \setminus X$.*

PROOF. Our proof is a minor modification of the proof of Lemma 14, [36]. Let T be a clique-tree of $G \setminus D$. Following [36], we say that a vertex v covers a bag x of T if v is contained in the bag x , and then T_v denotes the sub-clique-tree of all the bags which are covered by v . Now, since Q and K are cliques in $G \setminus D$, there are two bags x and y which contain Q and K , respectively. Consider the unique path connecting x and y in T , and suppose that the bags on this path are numbered as $x = 1, 2, \dots, s = y$. Let u_1, u_2, \dots be vertices of K with label t , and let a_i denote the

smallest numbered bag on this path that occurs in T_{u_i} , i.e. the smallest numbered bag containing u_i . Similarly, let v_1, v_2, \dots be the vertices of Q and let b_i denote the largest numbered bag which is contained in T_{v_i} . It follows that T_{v_i} and T_{u_j} intersect if and only if $a_j \leq b_i$, in which case there is an edge (v_i, u_j) in the graph. We also assume that the collection of a_i and b_i are distinct, which is easily achieved by adding additional bags along this path in T . Further we assume that the vertices u_1, u_2, \dots and v_1, v_2, \dots are ordered so that the sequences of a_i and b_j are strictly increasing.

We define a subsequence of b_i and a_j as follows. Let $\beta_1 = 1$, and for ever $j \geq 1$ let α_j be the smallest value such that $a_{\alpha_j} > b_{\beta_j}$. For every $i \geq 2$, let β_i be the smallest value such that $b_{\beta_i} > a_{\alpha_{i-1}}$. Observe that we obtain a strictly increasing sequence, $b_{\beta_1} < a_{\alpha_1} < b_{\beta_2} < a_{\alpha_2} \dots$. Let β_ℓ be the last element of the above sequence which corresponds to a vertex in Q .

Now, let u_s be a witness of a t^* -dangerous vertex v_{β_j} . We will show that a_{α_j} is also a witness for v_{β_j} . Clearly, $a_s > b_{\beta_j}$, which implies $a_s \geq a_{\alpha_j}$. Hence, the t^* -witness path from v_{β_j} to u_s passes through the bag a_{α_j} which contains u_{α_j} (see the proof of Lemma 14, [36]). Further, $a_{\alpha_j} > b_{\beta_j}$ implies that v_{β_j} and a_{α_j} are not neighbors in G . Therefore, a_{α_j} is also a witness for v_{β_j} .

Now, suppose that $\ell \leq (k+2)^2$. Then for each $i = 1, 2, \dots, \ell$ we mark the $k+2$ vertices $u_{\alpha_i}, u_{\alpha_i+1}, \dots, u_{\alpha_i+k+1}$ (if they exist), and we will show that this set of marked vertices contains a sufficient number of witnesses for every vertex in Q . Note that we have marked at most $(k+2)^3$ vertices of K . Consider any set X of k vertices such that a vertex $v_x \in Q$ has a witness path in $G \setminus X$ to some unmarked $u_y \in K$. In other words there is a chordless cycle in $G \setminus X$ which includes the vertex t , a t^* -dangerous vertex v_x and its t^* -witness $u_y \in K$. Since v_x and u_y are non-neighbors, we have that $b_x < a_y$, which implies that there is some j for which $b_x < a_{\alpha_j} \leq a_y$. If a_y is not marked then $y > \alpha_j + k + 1$, and hence $G \setminus X$ contains some $b_x < a_{\alpha_{j+r}} < a_y$. Now the witness path from v_x to a_y must contain a neighbor of $a_{\alpha_{j+r}}$, which implies that $a_{\alpha_{j+r}}$ is also a witness for v_x , in $G \setminus X$.

Finally, suppose that $\ell > (k+2)^2$. We will show that either the vertex t must be part of any solution of size k , or we can find a new mandatory edge in the graph. Let P_i be a witness path from v_{β_i} to u_{α_i} for every $1 \leq i \leq \ell$. Consider the collection of chordless cycles $H_{(k+2)-j}$ for $1 \leq j \leq k+1$, where $H_{(k+2)-j} = (t v_{\beta_{(k+2)-j}} P_{(k+2)-j} u_{\alpha_{(k+2)-j}} t)$. If all these cycles are pairwise vertex disjoint except for the vertex t , then we have obtained a collection of $k+1$ chordless cycles with t as the only common vertex. Hence t must be part of any solution of size at most k .

Otherwise, let $H_{(k+2)-j}$ and $H_{(k+2)-j'}$ have a common vertex w , for some $j < j'$. Then observe that w can only be an internal vertex of the paths $P_{(k+2)-j}$ and $P_{(k+2)-j'}$, and therefore it is not a neighbor of t . Since T is a tree-decomposition, this implies that w occurs in every bag numbered between $a_{\alpha_{(k+2)-j}}$ and $b_{\beta_{(k+2)-j'}}$. Since $a_{\alpha_{(k+2)-j}} < b_{\beta_{(k+2)-j+1}} < a_{\alpha_{(k+2)-j+1}} < \dots < b_{\beta_{(k+2)-j+k+1}} < a_{\alpha_{(k+2)-j+k+1}} < b_{\beta_{(k+2)-j'}}$, we have that w occurs in every one of these bags. Observe that we obtain a collection of $k+1$ chordless cycles of length 4, namely $(t v_{\beta_{(k+2)-j+i}} w u_{\alpha_{(k+2)-j+i}} t)$ for $i = 1, 2, \dots, k+1$, such that the vertices t and w are the only common vertices. Since t and w are non-adjacent in G , we obtain a new mandatory edge (t, w) in G . \square

Since $G \setminus D$ is a chordal graph which contains all the dangerous vertices for K , it follows that the graph induced by the set of all t -dangerous (t^* -dangerous) vertices forms a chordal graph as well. Since a chordal graph is perfect, it has a clique cover of size α , where α is the cardinality of a maximum independent set in the graph (see [23]). This gives us the following lemma, using the bound on the size of an independent set of dangerous vertices.

LEMMA 4.8 (LEMMAS 15 & 16, [36]). (i) *Either we can find a vertex that must be part of any solution of size at most k , or we can find a new mandatory edge, or we can mark $6k^2(k+1)$ vertices in K such that for any set X of k vertices, if a t -dangerous vertex v has a unmarked witness in $K \setminus X$, then it also has a marked witness in $K \setminus X$.*

- (ii) Either we can find a vertex that must be part of any solution of size at most k , or we can find a new mandatory edge, or we can mark $\Delta \cdot (k + 2)^3$ vertices in K such that for any set X of k vertices, if a t^* -dangerous vertex has a unmarked witness in $K \setminus X$, then it also has a marked witness in $K \setminus X$.

From the above lemma we conclude that, for any set X of at most k vertices, to test if X intersects all those chordless cycles that pass through t , K and some dangerous vertex v , it is sufficient to consider only the marked vertices of K as witnesses. However the above lemma considers only a single vertex $t \in D$, and we must mark additional witness vertices for each $t \in D$.

- LEMMA 4.9 (LEMMA 15 & 16, [36]). (i) Either we can find a vertex that must be part of any solution of size at most k , or we can find a new mandatory edge, or we can mark $|D| \cdot 6k^2(k + 1)$ vertices in K such that for any set X of k vertices, if a t -dangerous vertex v has a unmarked witness in $K \setminus X$, then it also has a marked witness in $K \setminus X$.
- (ii) Either we can find a vertex that must be part of any solution of size at most k , or we can find a new mandatory edge, or we can mark $|D| \cdot \Delta \cdot (k + 2)^3$ vertices in K such that for any set X of k vertices, if a t^* -dangerous vertex has a unmarked witness in $K \setminus X$, then it also has a marked witness in $K \setminus X$.

4.2 Fragments of Chordless cycles intersecting K

Now we shall mark vertices for chordless cycles in G that intersect K . If H is a chordless cycle in G , then consider F, P_1, P_2, \dots, P_s where $F = H \cap D$ and P_1, P_2, \dots, P_s are the paths in $H \setminus D$. It follows that each P_i has exactly two labels from F on its endpoints, and the internal vertices have no labels from F , and further these paths are pairwise independent (i.e. there is no edge between two vertices that are in two different paths). We call F, P_1, \dots, P_s the *fragments* of the chordless cycle H . Since K is a clique, at most one of these paths intersects K , which we assume to be the path P_1 , and further it contains at most two vertices from K . We will show that we can mark a bounded number of vertices in K such that for any chordless cycle H that includes an unmarked vertex (that lies in P_1), there is another chordless cycle H' that avoids this unmarked vertex. Let us first consider the case when P_1 just a single vertex (in K). The following lemma is a close variant of Lemma 18, [36] for this case.

LEMMA 4.10 (LEMMA 18, [36]). Let F, P_1, \dots, P_s be the fragments of H where P_1 is just a single vertex that lies in K . Then either we can find a new mandatory edge, or we can mark $|D|^3 \cdot (k + 2)$ vertices in K such that the following holds. For any set X of k vertices such that $G \setminus X$ has a chordless cycle which intersects K in an unmarked vertex, there is another chordless cycle in $G \setminus X$ which does not use any unmarked vertices of K .

PROOF. Our proof of this lemma is obtained by modifying proof of Lemma 18, [36]. And note that, as P_1 is a single vertex, F must have at least two vertices. For every $l_1, l_2, l_3 \in D$, mark $k + 1$ vertices of K which have labels l_1, l_2 and not l_3 . Hence, in total we have marked $|D|^3 \cdot (k + 1)$ vertices of K . The results of [36] shows that this is sufficient for the case when F has 3 or more vertices.

When F contains only two vertices l_1, l_2 , we need to mark some additional vertices. Let us recall that P_1 is only a single vertex of K and has the labels l_1, l_2 , and so it follows that l_1, l_2 are non-neighbors. Let x be the bag in the clique-tree of $G \setminus D$, which corresponds to the maximal clique K , and further assume that x is the root of this tree-decomposition. For any chordless cycle H^i such that $H^i \setminus D = P_1^i, P_2^i$ where P_1^i is a single vertex, let w_i be the bag closest to x in the tree decomposition that contains a vertex of P_2^i . Note that $w_i \neq x$, as vertices of P_1^i and P_2^i have no edges between them, and it follows that the vertices of P_2^i are contained in the sub-clique-tree rooted at w_i , and furthermore none of them are present in any bag outside this sub-clique-tree. We can

generate a list of these bags w_1, w_2, \dots , by considering every choice of P_1^i, l_1 and l_2 and selecting those bags of the tree decomposition that have a path from l_1 to l_2 with at least one internal vertex in the non-neighborhood of the choice of P_1^i . Among these bags, select $w_{i_1}, w_{i_2}, \dots, w_{i_q}$ such that none of its descendants in the clique-tree are in the collection w_1, w_2, \dots computed above. Note that, by definition no bag selected above is an ancestor or a descendant of another selected bag. Let $H_{i_1}, H_{i_2}, \dots, H_{i_q}$ be a collection of chordless cycles such that $P_2^{i_j}$ is contained in the sub-clique-tree of w_{i_j} , for $j = 1, 2, \dots, q$. Note that the collection of $P_2^{i_j}$ are pairwise vertex disjoint, and further there are no edges between the vertices of $P_2^{i_j}$ and $P_2^{i_{j'}}$ for $j \neq j'$. This follows from the fact that no bag outside the sub-clique-tree rooted at w_{i_j} contains any vertex of $P_2^{i_j}$, and that no bag selected in the collection of w_{i_j} is an ancestor or a descendant of another.

Consider the case when $q \leq k + 1$. We define the distance of a vertex v from a bag w in the clique-tree as the minimum of the distance between w and a bag x that contains v . Then for each w_{i_j} , sort the vertices of K , which have labels l_1, l_2 , according to the distance from w_{i_j} , and mark $k + 1$ vertices from w_{i_j} . It follows that we mark $|D|^2 \cdot (k + 1)^2$ vertices of K . Let us argue that these marked vertices satisfy the requirements. For any set X of at most k vertices, suppose H is a chordless cycle in $G \setminus X$ with fragments F, P_1, P_2 where P_1 is just a single vertex $u \in K$ with labels $l_1, l_2 \in F$. Consider some bag w_i whose sub-clique-tree contains the path P_2 , and note that w_i , or some descendent w_{i_j} was selected in the above collection. Now if u was not marked for w_{i_j}, l_1, l_2 , then any of the vertices that were marked for this tuple has a greater distance from the bags w_{i_j} (and w_i) than u . Since we marked $k + 1$ vertices for this tuple, at least one of them is present in $K \setminus X$ and let u' be that vertex. It follows that replacing u with u' in H gives us a chordless cycle in $G \setminus X$.

Now, consider the case when $q \geq k + 2$. The collection of paths P^{i_j} , along with l_1 and l_2 , defines a collection of chordless cycles such that any solution of size k must pick at least one of l_1 or l_2 . Since l_1, l_2 are non-neighbors, we obtain a new mandatory edge (l_1, l_2) .

Further note that the total number of marked vertices is $|D|^3 \cdot (k + 1) + |D|^2 \cdot (k + 1)^2$. Since $|D| \geq k$, it follows that the total number of marked vertices is upper bounded by $|D|^3 \cdot (k + 2)$. \square

Now, the only remaining case is when the path P_1 has two or more vertices. For this case, we have the following lemma, which follows from Lemma 19 of [36] where we rely on Lemma 4.9.

LEMMA 4.11 (LEMMA 19, [36]). *Let F, P_1, \dots, P_s be the fragments of a chordless cycle H where P_1 contains at least two vertices, and further at least one of those vertices are in K . Then we can mark at most $|D| \cdot 6k^2(k + 1) + |D| \cdot \Delta \cdot (k + 2)^3$ vertices in K such that, for any set X of at most k vertices, if $G \setminus X$ contains a chordless cycle that includes an unmarked vertex $u \in K$, then $G \setminus X$ also contains a chordless cycle that avoids u , and it has strictly fewer unmarked vertices of K .*

Combining all of the above lemmas we obtain the following.

LEMMA 4.12. *Let (G, k, E_M, E_I) be an instance of CVD whose independent-degree is bounded by Δ . Let D be solution to this instance and let K be a maximal clique in $G \setminus D$. Then in polynomial time, either we can find a vertex which must be part of any solution of size at most k , or we can find a new mandatory edge, or we can safely remove all but $c \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k + 2)^3)$ vertices of K . (Here c is some constant independent of the input.)*

PROOF. We apply the Lemmas 4.9, 4.10 and 4.11 to the given instance. If any of them return a vertex that must be part of any solution of size at most k , or a new mandatory edge then we output that vertex or edge. Otherwise, together they mark a maximum of $c' \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k + 2)^3)$ vertices in K , where c' is some constant independent of the input. In addition, let us also mark any

unmarked vertex in K which is an endpoint of a mandatory edge in the instance. Since the total number of mandatory edges is always bounded by k^2 , the total number of marked vertices in K does not exceed $c \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k+2)^3)$, where c is again a constant independent of the input.

Let u be any unmarked vertex in K . We will show that the instances G and $G \setminus u$ are equivalent. Let X be any set of at most k vertices such that $(G \setminus u) \setminus X$ is a chordal graph. If $G \setminus X$ is not chordal, then there is some chordless cycle H in $G \setminus X$. If H does not contain the vertex u , then it is also present in $(G \setminus u) \setminus X$ which is a contradiction. On the other hand if H contains u , then by the above lemmas, we can argue that there is some chordless cycle H' in $G \setminus X$ which avoids u (see Lemma 19, [36]). Therefore H' is present in $(G \setminus u) \setminus X$, which is again a contradiction. Now, by induction, we can show that it is safe to remove all the unmarked vertices in K from the graph. \square

Now we are ready to prove Lemma 4.1. Consider any maximal clique in the graph $G \setminus D$ with more than $c \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k+2)^3)$ vertices, and apply Lemma 4.12 to it. If it returns a vertex v that must be part of any solution of size k , we remove it from the graph and decrease k by one. Else if it returns a mandatory edge, we add this edge to the graph and mark it as mandatory. We can argue, as before, that both these operations are safe. We also apply Reduction Rule 2 to the instance if necessary, to bound the number of mandatory edges in the graph. Otherwise Lemma 4.12 ensures that it is safe to remove all but the $c \cdot (|D|^3 \cdot k + |D| \cdot \Delta \cdot (k+2)^3)$ marked vertices of K . Therefore, we remove these vertices from the graph. We also update the sets E_M and E_I as follows. Note that both the endpoints of every mandatory edge (in E_M) are marked by the Lemma 4.12, and hence they are not removed. However, we may need to update the collection of irrelevant edges, E_I , by removing those edges whose endpoints were deleted.

Observe that, each application of Lemma 4.12 either reduces k , or adds a new mandatory edge, or reduces the number of vertices in the graph. Further, we may add at most $k^2 + 1$ new mandatory edges before finding a new vertex that must be part of any solution of size k or concluding that the given instance is a No instance. Hence, in polynomial time, either we bound the size of every maximal clique in the graph or conclude that the given instance is a No instance.

5 CONCLUSION

In this paper we obtained a polynomial kernel for CVD of size $O(k^{12} \log^{10} k)$. The new kernel significantly improves over the previously known $O(k^{161} \log^{58} k)$ sized kernel. We believe that the notion of independence degree and the bootstrapping trick used in our kernelization procedure could be useful in designing polynomial kernels for other \mathcal{F} -VERTEX (EDGE) DELETION problems, where \mathcal{F} is characterized by an infinite set of forbidden induced graphs. We conclude the paper with the following open problems.

- Design a polynomial kernel for CVD of size $O(k^c)$ for some fixed constant $c < 10$.
- Design a constant-factor approximation algorithm for CVD.
- Does there exist an FPT algorithm for CVD with running time $c^k n^{O(1)}$, for some fixed constant c ?

REFERENCES

- [1] Faisal N. Abu-Khzam. 2010. A kernelization algorithm for d -Hitting Set. *J. Comput. Syst. Sci.* 76, 7 (2010), 524–531.
- [2] Faisal N. Abu-Khzam, Rebecca L. Collins, Michael R. Fellows, Michael A. Langston, W. Henry Suters, and Christopher T. Symons. 2004. Kernelization Algorithms for the Vertex Cover Problem: Theory and Experiments. In *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics, New Orleans, LA, USA, January 10, 2004*. 62–69.
- [3] Akanksha Agrawal, Daniel Lokshantov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. 2017. Polylogarithmic Approximation Algorithms for Weighted- \mathcal{F} -Deletion Problems. *Manuscript available upon request* (2017).

- [4] Noga Alon, Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. 2011. Solving MAX- r -SAT Above a Tight Lower Bound. *Algorithmica* 61, 3 (2011), 638–655.
- [5] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. 2009. On problems without polynomial kernels. *J. Comput. Syst. Sci.* 75, 8 (2009), 423–434. <https://doi.org/10.1016/j.jcss.2009.04.001>
- [6] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. 2016. (Meta) Kernelization. *J. ACM* 63, 5 (2016), 44:1–44:69. <http://dl.acm.org/citation.cfm?id=2973749>
- [7] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. 2013. Preprocessing for Treewidth: A Combinatorial Analysis through Kernelization. *SIAM J. Discrete Math.* 27, 4 (2013), 2108–2142. <https://doi.org/10.1137/120903518> arXiv:1104.4217
- [8] Yixin Cao. 2016. Linear Recognition of Almost Interval Graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. 1096–1115.
- [9] Yixin Cao and Dániel Marx. 2015. Interval Deletion Is Fixed-Parameter Tractable. *ACM Transactions on Algorithms* 11, 3 (2015), 21:1–21:35.
- [10] Yixin Cao and Dániel Marx. 2016. Chordal Editing is Fixed-Parameter Tractable. *Algorithmica* 75, 1 (2016), 118–137.
- [11] Jianer Chen, Iyad A. Kanj, and Weijia Jia. 2001. Vertex cover: further observations and further improvements. *Journal of Algorithms* 41, 2 (2001), 280–301.
- [12] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer-Verlag.
- [13] Marek Cygan, Lukasz Kowalik, and Marcin Pilipczuk. 2013. Open problems from workshop on kernels. URL: <http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf> (2013).
- [14] Holger Dell and Dieter van Melkebeek. 2014. Satisfiability Allows No Nontrivial Sparsification unless the Polynomial-Time Hierarchy Collapses. *J. ACM* 61, 4 (2014), 23. <https://doi.org/10.1145/2629620>
- [15] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer.
- [16] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer-Verlag, Berlin. 493 pages.
- [17] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. 2012. Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*. 470–479. <https://doi.org/10.1109/FOCS.2012.62>
- [18] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. 2010. Bidimensionality and Kernels. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*. 503–510.
- [19] Fedor V. Fomin, Saket Saurabh, and Yngve Villanger. 2013. A Polynomial Kernel for Proper Interval Vertex Deletion. *SIAM J. Discrete Math.* 27, 4 (2013), 1964–1976.
- [20] Lance Fortnow and Rahul Santhanam. 2011. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.* 77, 1 (2011), 91–106. <https://doi.org/10.1016/j.jcss.2010.06.007>
- [21] Toshihiro Fujito. 1998. A Unified Approximation Algorithm for Node-deletion Problems. *Discrete Applied Mathematics* 86, 2-3 (1998), 213–231. [https://doi.org/10.1016/S0166-218X\(98\)00035-3](https://doi.org/10.1016/S0166-218X(98)00035-3)
- [22] Archontia C. Giannopoulou, Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. 2015. Uniform Kernelization Complexity of Hitting Forbidden Minors. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*. 629–641. https://doi.org/10.1007/978-3-662-47672-7_51
- [23] Martin Charles Golumbic. 1980. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York.
- [24] Pinar Heggenes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. 2013. Parameterized complexity of vertex deletion into perfect graph classes. *Theor. Comput. Sci.* 511 (2013), 172–180.
- [25] Bart M. P. Jansen. 2014. Turing Kernelization for Finding Long Paths and Cycles in Restricted Graph Classes. In *Proc. 22th ESA (Lecture Notes in Computer Science)*, Andreas S. Schulz and Dorothea Wagner (Eds.), Vol. 8737. Springer, 579–591. https://doi.org/10.1007/978-3-662-44777-2_48
- [26] Bart M. P. Jansen and Marcin Pilipczuk. 2017. Approximation and Kernelization for Chordal Vertex Deletion. In *Proceedings of the Twenty-Eight Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*. to appear.
- [27] Yuping Ke, Yixin Cao, Xiating Ouyang, and Jianxin Wang. 2016. Unit Interval Vertex Deletion: Fewer Vertices are Relevant. *arXiv preprint arXiv:1607.01162* (2016).
- [28] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. 2016. Linear Kernels and Single-Exponential Algorithms Via Protrusion Decompositions. *ACM Trans. Algorithms* 12, 2 (2016), 21. <https://doi.org/10.1145/2797140>
- [29] Stefan Kratsch. 2012. Polynomial Kernelizations For $\text{MIN } F^+ \Pi_1$ And MAX NP. *Algorithmica* 63, 1-2 (2012), 532–550. <https://doi.org/10.1007/s00453-011-9559-5>
- [30] Stefan Kratsch. 2014. Recent developments in kernelization: A survey. *Bulletin of the EATCS* 113 (2014). <http://eatcs.org/beatcs/index.php/beatcs/article/view/285>

- [31] Stefan Kratsch and Magnus Wahlström. 2012. Representative Sets and Irrelevant Vertices: New Tools for Kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*. 450–459. <https://doi.org/10.1109/FOCS.2012.46>
- [32] Stefan Kratsch and Magnus Wahlström. 2014. Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal. *ACM Trans. Algorithms* 10, 4 (2014), 20:1–20:15. <https://doi.org/10.1145/2635810>
- [33] John M. Lewis and Mihalis Yannakakis. 1980. The Node-Deletion Problem for Hereditary Properties is NP-Complete. *J. Comput. Syst. Sci.* 20, 2 (1980), 219–230.
- [34] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. 2012. Kernelization - Preprocessing with a Guarantee. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*. 129–161.
- [35] Carsten Lund and Mihalis Yannakakis. 1994. On the Hardness of Approximating Minimization Problems. *J. ACM* 41, 5 (1994), 960–981. <https://doi.org/10.1145/185675.306789>
- [36] Dániel Marx. 2010. Chordal Deletion is Fixed-Parameter Tractable. *Algorithmica* 57, 4 (2010), 747–768.
- [37] Dániel Marx, Barry O’Sullivan, and Igor Razgon. 2013. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms* 9, 4 (2013), 30.
- [38] Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications, Vol. 31. Oxford University Press, Oxford. xii+300 pages.
- [39] Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. 2014. Network Sparsification for Steiner Problems on Planar and Bounded-Genus Graphs. In *Proc. 55th FOCS*. IEEE Computer Society, 276–285. <https://doi.org/10.1109/FOCS.2014.37>
- [40] Stéphan Thomassé. 2010. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms* 6, 2 (2010), 32:1–32:8. <https://doi.org/10.1145/1721837.1721848>
- [41] Mihalis Yannakakis. 1978. Node- and Edge-Deletion NP-Complete Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*. 253–264. <https://doi.org/10.1145/800133.804355>

Received February 2007; revised March 2009; accepted June 2009; revised January 2017