Interval Vertex Deletion Admits a Polynomial Kernel*

Akanksha Agrawal† Pranabendu Misra‡ Saket Saurabh§ Meirav Zehavi¶

Abstract

Given a graph G and an integer k, the Interval Vertex Deletion (IVD) problem asks whether there exists a subset $S \subseteq V(G)$ of size at most k such that $G - S$ is an interval graph. This problem is known to be NP-complete [Yannakakis, STOC'78]. Originally in 2012, Cao and Marx showed that IVD is fixed parameter tractable: they exhibited an algorithm with running time $10^k n^{\mathcal{O}(1)}$ [Cao and Marx, SODA'14]. The existence of a polynomial kernel for IVD remained a well-known open problem in Parameterized Complexity. In this paper, we settle this problem in the affirmative.

1 Introduction

In a *graph modification problem*, the input consists of an n-vertex graph G and an integer k. The objective is to determine whether k *modification operations*—such as vertex deletions, or edge deletions, insertions or contractions—are sufficient to obtain a graph with prescribed structural properties such as being planar, bipartite, chordal, interval, acyclic or edgeless. Graph modification problems include some of the most basic problems in graph theory and graph algorithms. Unfortunately, most of these problems are NP-complete [43, 50]. Therefore, they have been studied intensively within algorithmic paradigms for coping with NP-completeness [21, 25, 46], including approximation algorithms, parameterized complexity, and algorithms for restricted input classes.

Graph modification problems have played a central role in the development of parameterized complexity, see the related works subsection. Here, the number of allowed modifications, k, is considered a *parameter*. With respect to k, we seek a *fixed parameter tractable (FPT)* algorithm, namely, an algorithm whose running time has the form $f(k)n^{\mathcal{O}(1)}$ for some computable function f. One way to obtain such an algorithm is to exhibit a *kernelization algorithm*, or *kernel*. A kernel for a graph problem Π is an algorithm that given an instance (G, k) of Π, runs in polynomial time and outputs an equivalent instance (G', k') of Π such that $|V(G')|$ and k' are upper bounded by $f(k)$ for some computable function f. The function f is called the *size* of the kernel, and if f is a polynomial function, then we say that the kernel is a *polynomial kernel*. A kernel for a (decidable) problem immediately implies that it admits an FPT algorithm, but kernels are also interesting in their own right. In particular, kernels allow us to model the performance of polynomial time pre-processing algorithms. The field of kernelization has received a significant amount of attention, especially after the introduction of methods for showing kernelization lower bounds [5, 14, 15, 18, 24, 29, 30]. We refer to the surveys [23, 28, 39, 44], as well as the books [12, 17, 19, 49], for a detailed treatment of the area of kernelization. In this paper, we study the kernelization complexity of the following problem.

Interval Vertex Deletion (IVD) **Input:** A graph G and an integer k. **Parameter:** k **Question:** Does there exist a subset $S \subseteq V(G)$ of size at most k such that $G - S$ is an interval graph?

A graph G is an *interval graph* if it is the intersection graph of intervals on the real line. Due to their intriguing combinatorial properties and many applications in diverse areas, such as industrial engineering and archeology [4, 36], the class of interval graphs is perhaps one of the most studied graph classes [7, 27]. Whether IVD admits an FPT algorithm has been a longstanding open problem in the area until it was resolved by Cao and Marx [10], who gave an algorithm with running time $\mathcal{O}(10^k n^9)$. Subsequently, Cao [9] designed an FPT algorithm with linear dependence on the input size, as well as slightly better dependence on the parameter k. More precisely, Cao's algorithm has running

*The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 306992.

†Hungarian Academy of Sciences, Budapest, Hungary. akanksha@sztaki.hu.

‡University of Bergen, Bergen, Norway. pranabendu.misra@uib.no.

§University of Bergen, Bergen, Norway. The Institute of Mathematical Sciences, HBNI, Chennai, India. UMI ReLax. saket@imsc.res.in.

¶Ben-Gurion University, Beersheba, Israel. meiravze@bgu.ac.il.

time $\mathcal{O}(8^k(n+m))$. A natural follow-up question to this work, explicitly asked multiple times in the literature [13, 31, 33], is whether IVD admits a polynomial kernel. In this paper, we resolve this question in the affirmative:

THEOREM 1.1. INTERVAL VERTEX DELETION *admits a polynomial kernel.*

1.1 Methods The first ingredient of our kernelization algorithm is the factor 8 polynomial time approximation algorithm for IVD by Cao [9]. We use this algorithm to obtain an approximate solution of size at most $8k$, or conclude that no solution of size at most k exists. By re-running the approximation algorithm on the graph with some of the vertices marked as "undeletable", we grow our approximate solution to a 9-*redundant* solution M of size $\mathcal{O}(k^{10})$. Here, 9-redundancy roughly means that for every subset $W \subseteq M$ of size at most 9, either $M \setminus W$ is also a solution, or every solution S' of size at most $k+2$ has non-empty intersection with W.[1]

Our kernelization heavily uses the characterization of interval graphs in terms of their *forbidden induced subgraphs*, also called *obstructions*. Specifically, a graph H is an obstruction to the class of interval graphs if H is not an interval graph, and for every vertex $v \in V(H)$ we have that $H - \{v\}$ is an interval graph. A graph G is an interval graph if and only if it does not contain any obstruction as an induced subgraph. The set of obstructions to interval graphs have been completely characterized by Lekkerkerker and Boland, [42]. It consists of the *long claw*, the *whipping top*, the *net*, the *tent*, as well as three infinite families of graphs: the *single-dagger asteroidal witness* (†-AW), the *double-dagger asteroidal witnesses* (‡-AW), and the cycle of length at least 4 (see Figure 1).

Having a 9-redundant solution yields the following advantage. In several places, we remove a carefully chosen vertex $v \notin M$ from G and claim that $G - \{v\}$ has a solution of size at most k if and only if G does. One direction of the equivalence is trivial. The interesting direction is to show that a solution X of size k to $G - \{v\}$ implies the existence of a solution of size at most k for G. The starting point for such an analysis is to ask why X is not already a solution for G. The only possible reason is that $G - X$ contains an obstruction $\mathbb{O}$, and $\mathbb{O}$ must contain v. We claim that $\mathbb{O}$ contains at least 10 vertices from M. Suppose not, then let W be the intersection of M and $\mathbb{O}$. We know that $(G - (M \setminus W))$ contains $\mathbb{O}$, and

[1]The precise definition in Section 3 contains another condition that is not specified in the introduction for the sake of clarity of exposition.

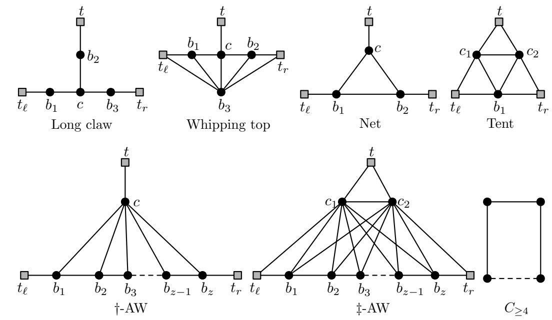

Figure 1: The set of obstructions for an interval graph.

therefore it is not an interval graph. Hence, by the 9-redundancy of M, this implies that X (being a solution of size at most $k+2$) must intersect $\mathbb{O}$, which contradicts the choice of $\mathbb{O}$. Thus, in this analysis we only need to care about *large* obstructions that, furthermore, have a large intersection with M. This is crucial throughout the design and analysis of the kernel.

We then proceed to classify the connected components of $G - M$ based on whether they are *modules* in G or not. (Recall that a module is a set X such that all vertices in X have the same neighbors outside X.) For each component C that is not a module, there is an edge (u, v) in C and a vertex w in M such that w is adjacent to u but not to v. Thus, if there are more than $(k+2)|M|$ non-module components in total, then there must exist $k+3$ non-module components and a vertex $w \in M$ such that each of these components has an edge (u, v) where w is adjacent to u but not to v. However, this means that for every subset $S \subseteq V(G)$ of size at most k, either $w \in S$ or $G - S$ contains a long claw (whose center c is w) and hence not interval. It follows that w must belong to every solution of size at most $k+2$; thus, we can simply remove w and decrease the budget k by 1. Hence, the number of non-module components can be bounded by $(k+2)|M|$, which is polynomial in k.

Since none of the obstructions contains any module on more than a single vertex, and the components of $G - M$ are interval graphs, it follows that every obstruction can intersect every module component in at most one vertex. Furthermore, there is no point in keeping more than $k+1$ copies of any vertex, so we can reduce the module components to cliques of size $k+1$.

We are left with the following situation. We have a 9-redundant solution M of size $\mathcal{O}(k^{10})$. At most $\mathcal{O}(|M|)$ components of $G - M$ are not modules, but these components could be arbitrarily large. The remaining components are all modules that are cliques of size at most $k+1$; thus, the module components are structured and small, but there could be arbitrarily many of them. This means that we are left with two tasks: *(i)* reduce the *number* of module components, and *(ii)* reduce the *size* of the non-module components. These two tasks can be approached separately, and both turn out to be non-trivial. Since both tasks are quite technically involved, we only give a few highlights in the remainder of this overview.

Bounding the Number of Module Components. Consider first the case where there are no non-module components at all, and every module component is a single vertex. In this case, $G - M$ is edgeless, so M is a *vertex cover* of G. The kernelization complexity of even this very special case was asked as an open problem by Fomin et al. [20].

A key ingredient in our solution to this special case is a new bound for the setting considered in the classic two families theorem of Bollobás [6]. Suppose there are two families of sets over a universe U, $A_1, \ldots, A_m$ and $B_1, \ldots, B_m$, such that every set A_i has size p, every set B_j has size q, for every i the sets A_i and B_i are disjoint, while for every $i \neq j$ the sets A_i and B_j intersect. The two families theorem gives an upper bound of $\binom{p+q}{p}$ for the size m of the family. The upper bound on m is *independent of the universe size*, and this has been extensively used in the design of parameterized algorithms [22, 47]. Further, when p or q is a *constant* the bound is *polynomial* in $p + q$, and this has been extensively used in kernelization [40].

In our setting neither the sets $A_1, \ldots, A_m$ nor the sets $B_1, \ldots, B_m$ have constant cardinality. However, we know that for every $i \neq j$, $|A_i \cap B_j| \in \{1, 2\}$. We prove that in this case, the bound is $\mathcal{O}(|U|^2)$. More generally, we prove the following.

Lemma 1.1. (Bounded Intersection Two Families) *Let $A_1, \ldots, A_m$ and $B_1, \ldots, B_m$ be families over a universe U such that (i) for every $i \leq m$, $A_i \cap B_i = \emptyset$, and (ii) for every $j \neq i$, $|A_i \cap B_j| \in \{1, \ldots, c\}$. Then $m \leq \sum_{t=0}^{c} \binom{|U|}{t}$.*

Comparing Lemma 1.1 with the Two Families Theorem, the bound in Lemma 1.1 does depend on the universe size $|U|$. On the other hand, the exponent of $|U|$ only depends on the maximum cardinality c of the *intersection* between the sets A_i and B_j.

In the setting of kernelizing IVD parameterized by the size of a vertex cover M, the size of the kernel is intimately linked to m for the case where $A_1, \ldots, A_m$ is a collection of cliques in $G[M]$ while $B_1, \ldots, B_m$ is a collection of induced paths. Since a clique can only intersect an induced path in at most two vertices, we can apply Lemma 1.1 with $c = 2$, thereby obtaining an $\mathcal{O}(|M|^2)$ bound for m and (after a significant amount of additional efforts, which we skip in this overview) a polynomial bound on the kernel size.

The kernel for IVD parameterized by vertex cover quite simply translates into a procedure that bounds the number, and therefore the total size, of module components of $G - M$. We remark that, because the *number* of non-module components is bounded by $\mathcal{O}(k|M|)$, by bounding the number of module components we also bound the total number of components of $G - M$.

Bounding the Size of Non-Module Components. Suppose now that the number of module components has been bounded by $k^{\mathcal{O}(1)}$. We can now include all of the module components in M, and proceed under the assumption that there are no module components at all.

The size-reduction of non-module components proceeds in three phases. In the first phase, we bound the maximum clique size in a component. Our clique-reduction procedure builds upon the clique-reduction procedure of Marx [48], which was used in kernelizations for Chordal Vertex Deletion [1, 34]. Both the procedure of Marx and ours are based on an "irrelevant vertex rule". However, our procedure is necessarily much more involved—our irrelevant vertex rule needs to preserve not only long induced cycles, but also large single and double dagger asteroidal witnesses.

Having reduced the maximum clique size in the component we proceed to the second phase, where we reduce the set of vertices that appear in at least two maximal cliques in the component. In this phase, we partition the component into $k^{O(1)}$ "long" and "thin" parts, and prove that an optimal solution will either not touch a part at all, or it will cut it into two pieces using a minimal separator. Then, provided that a part is sufficiently large, we identify an edge whose contraction does not decrease the size of any minimal separator inside the part. Thus, on the one hand, contracting e does not decrease the size of an optimal solution. On the other hand, contracting e—or any edge for that matter—cannot *increase* the size of an optimal solution (since interval graphs are closed under contraction).

After the second phase, the number of vertices appearing in at least two maximal cliques of the component is upper bounded by $k^{\mathcal{O}(1)}$. In the third phase, we bound the number of the remaining vertices—these are the vertices that are "private" to some maximal clique of the component. At this point we can take the set of vertices appearing in at least two components and add them to M. This makes M grow by $k^{\mathcal{O}(1)}$ vertices, but now the large component breaks up into components whose size is not larger than that of a maximal clique, that is, $k^{\mathcal{O}(1)}$. We can now re-apply the procedure for bounding the number of components and this bounds the total number of vertices in G by $k^{\mathcal{O}(1)}$. *We remark that, for technical reasons, in the actual proof phases* 2 *and* 3 *as described here are interleaved.*

1.2 Related Works on Parameterized Graph Modification Problems The $\mathcal{F}$-Vertex Deletion problems corresponding to the families of edgeless graphs, forests, chordal graphs, interval graphs, bipartite graphs, and planar graphs are known as Vertex Cover, Feedback Vertex Set, Chordal Vertex Deletion, Interval Vertex Deletion, Odd Cycle Transversal/Vertex Bipartization and Planar Vertex Deletion, respectively. These problems are among the most well studied problems in the field of parameterized complexity. The study of parameterized graph deletion problems together with their various restrictions and generalizations has been an extremely active subarea over the last few years. In fact, just over the course of the last few years there have been results on parameterized algorithms for Chordal Editing [11], Unit Interval Vertex (Edge) Deletion [8, 35], Interval Vertex (Edge) Deletion [9, 10], Planar $\mathcal{F}$ Deletion [21, 38], Planar Vertex Deletion [32], Block Graph Deletion [37] and Simultaneous Feedback Vertex Set [3]. It is important to note that for many of these problems, polynomial kernels gave rise to several new techniques in the area. However, the problem which is closest to ours is the Chordal Vertex Deletion problems. In a recent breakthrough, Jansen and Pilipczuk [34] gave a polynomial kernel (of size $\mathcal{O}(k^{162})$) for Chordal Vertex Deletion, resolving a more than a decade old open problem. Shortly afterwards, Agrawal et al. [1, 2] gave a kernel of size $\mathcal{O}(k^{12}\log^{10} k)$.

2 Preliminaries

We denote the set of natural numbers by $\mathbb{N}$. For $n \in \mathbb{N}$, we use $[n]$ and $[n]_0$ as shorthands for $\{1, 2, \ldots, n\}$ and $\{0, 1, \ldots, n\}$, respectively. For a set X and an integer $n \in \mathbb{N}$, by X^n we denote the set $\{(a_1, a_2, \ldots, a_n) \mid a_1, a_2, \ldots, a_n \in X\}$.

Basic Graph Theory. We refer to standard terminology from the book of Diestel [16] for those graph-related terms that are not explicitly defined here. Given a graph G, we denote its vertex set and its edge set by $V(G)$ and $E(G)$, respectively. Given a set $\mathcal{C}$ of connected components of G, denote $V(\mathcal{C}) = \bigcup_{C\in\mathcal{C}} V(C)$. Moreover, when the graph G is clear from context, denote $n = |V(G)|$. Given a subset $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G induced by U. Accordingly, a graph H is an *induced subgraph* of G if there exists $U \subseteq V(G)$ such that $G[U] = H$. For a set of vertices $X \subseteq V(G)$, $G - X$ denotes the induced subgraph $G[V(G) \setminus X]$, i.e. the graph obtained by deleting the vertices in X from G. For an edge $(u, v) \in E(G)$, $G/(u, v)$ denotes the graph obtained by contracting the edge (u, v), i.e. the graph obtained by introducing a new vertex that is adjacent to all vertices in $(N(u) \cup N(v)) \setminus \{u, v\}$ and deleting the vertices $\{u, v\}$. We say that G is a *clique* if for all distinct vertices $u, v \in V(G)$, we have that $(u, v) \in E(G)$, and that $V(G)$ is an independent set if for all distinct vertices $u, v \in V(G)$, we have that $(u, v) \notin E(G)$. Given a vertex $v \in V(G)$, $N_G(v)$ denotes the neighborhood of v in G. Moreover, a subset $U \subseteq V(G)$ is a *module* if for all $u, u' \in U$ and $v \in V(G) \setminus U$, either both u and u' are adjacent to v or both u and u' are not adjacent to v. For the sake of simplicity, we also call $G[U]$ a module.

A *path* $P = (x_1, x_2, \ldots, x_\ell)$ in G is a subgraph of G where $V(P) = \{x_1, x_2, \ldots, x_\ell\} \subseteq V(G)$ and $E(P) = \{(x_i, x_{i+1}) \mid i \in [\ell - 1]\} \subseteq E(G)$, where $\ell \in [n]$. The vertices x_1 and x_ℓ are the *endpoints* of P, and the remaining vertices in $V(P)$ are the *internal vertices* of P. A *cycle* $C = (x_1, x_2, \ldots, x_\ell)$ in G is a subgraph of G where $V(C) = \{x_1, x_2, \ldots, x_\ell\} \subseteq V(G)$ and $E(C) = \{(x_i, x_{i+1}) \mid i \in [\ell-1]\} \cup \{(x_1, x_\ell)\} \subseteq E(G)$. We say that $(u, v) \in E(G)$ is a *chord* of a path P if $u, v \in V(P)$ but $(u, v) \notin E(P)$. Similarly, we say that $(u, v) \in E(G)$ is a *chord* of a cycle C if $u, v \in V(C)$ but $(u, v) \notin E(C)$. A path P or cycle C is said to be *induced* (or, alternatively, *chordless*) if it has no chords.

Interval Graphs. An *interval graph* is a graph that does not contain any of the following graphs, called *obstructions*, as an induced subgraph (see Figure 1).

- **Long Claw.** A graph $\mathbb{O}$ such that $V(\mathbb{O}) = \{t_\ell, t_r, t, c, b_1, b_2, b_3\}$ and $E(\mathbb{O}) = \{(t_\ell, b_1), (t_r, b_3), (t, b_2), (c, b_1), (c, b_2), (c, b_3)\}$.
- **Whipping Top.** A graph $\mathbb{O}$ such that $V(\mathbb{O}) = \{t_\ell, t_r, t, c, b_1, b_2, b_3\}$ and $E(\mathbb{O}) = \{(t_\ell, b_1), (t_r, b_2), (c, t), (c, b_1), (c, b_2), (b_3, t_\ell), (b_3, b_1), (b_3, c), (b_3, b_2), (b_3, t_r)\}$.
- **†-AW.** A graph $\mathbb{O}$ such that $V(\mathbb{O}) = \{t_\ell, t_r, t, c\} \cup \{b_1, b_2, \ldots, b_z\}$, where $t_\ell = b_0$ and $t_r = b_{z+1}$, $E(\mathbb{O}) = \{(t, c), (t_\ell, b_1), (t_r, b_z)\} \cup \{(c, b_i) \mid i \in [z]\} \cup \{(b_i, b_{i+1}) \mid i \in [z - 1]\}$, and $z \geq 2$. A †-AW where $z = 2$ will be called a *net*.
- **‡-AW.** A graph $\mathbb{O}$ such that $V(\mathbb{O}) = \{t_\ell, t_r, t, c_1, c_2\} \cup \{b_1, b_2, \ldots, b_z\}$, where $t_\ell = b_0$ and $t_r = b_{z+1}$, $E(\mathbb{O}) = \{(t, c_1), (t, c_2), (c_1, c_2), (t_\ell, b_1), (t_r, b_z), (t_\ell, c_1), (t_r, c_2)\} \cup \{(c, b_i) \mid i \in [z]\} \cup \{(b_i, b_{i+1}) \mid i \in [z - 1]\}$, and $z \geq 1$. A ‡-AW where $z = 1$ will be called a *tent*.
- **Hole.** A chordless cycle on at least four vertices.

An obstruction $\mathbb{O}$ is *minimal* if there does not exist an obstruction $\mathbb{O}'$ such that $V(\mathbb{O}') \subset V(\mathbb{O})$. We refer to †-AW and ‡-AW as AWs. In each of the first four obstructions, the vertices t_ℓ, t_r, and t are called *terminals*, the vertices c, c_1, and c_2 are called *centers*, and the other vertices are called *base vertices*. Furthermore, the vertex t is called the *shallow terminal* and the vertices t_ℓ and t_r are called the *non-shallow terminals*. In the case where $\mathbb{O}$ is one of the AWs, the induced path on the set of base vertices is called the *base* of the AW, and it is denoted by $\mathsf{base}(\mathbb{O})$. Moreover, we say that the induced path on the set of base vertices, t_ℓ and t_r is the *extended base* of the AW, and it is denoted by $P(\mathbb{O})$.

Path Decomposition. A *path decomposition* of a connected graph G is a pair (P, β) where P is a path, and $\beta : V(P) \rightarrow 2^{V(G)}$ is a function that satisfies the following properties.

(i) $\bigcup_{x \in V(P)} \beta(x) = V(G)$,

(ii) For any edge $(u, v) \in E(G)$ there is a node $x \in V(P)$ such that $u, v \in \beta(x)$.

(iii) For any $v \in V(G)$, the collection of nodes $P_v = \{x \in V(P) \mid v \in \beta(x)\}$ is a subpath of P.

For $v \in V(P)$, we call $\beta(v)$ the *bag* of v. We refer to the vertices in $V(P)$ as nodes. A *clique path* of a connected graph G is a path decomposition of G where every bag is a distinct maximal clique. If a graph G admits a clique path, then we say that G is a clique path. The following proposition states that the class of interval graphs is exactly the class of graphs where each connected component is a clique path.

Proposition 2.1. ([26, 27]) *A graph is an interval graph if and only if each connected component of it is a clique path.*

Parameterized Complexity. Let Π be an NP-hard problem. In the framework of Parameterized Complexity, each instance of Π is associated with an integer k, which is called the *parameter*. Here, the goal is to confine the combinatorial explosion in the running time of an algorithm for Π to depend only on k. The main concepts defined to achieve this goal are of *fixed-parameter tractability* and *kernelization*. First, we say that Π is *fixed-parameter tractable (FPT)* if any instance (I, k) of Π is solvable in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, where $f(\cdot)$ is an arbitrary (computable) function of k. Second, Π 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 transforms the input instance into an equivalent instance of Π 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 Π. For a detailed introduction to the field of kernelization, we refer to the following surveys [39, 44] and the corresponding chapters in the books [12, 17, 19, 49].

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 *safe* if the instances (I, k) and (I', k') are equivalent.

Linear Algebra. For a set A and X, by an *operation of A onto X* we mean a function $f : A \times X \rightarrow X$. For an element $(a, x) \in A \times X$ by ax we denote the element $f(a, x) \in X$. For a field $\mathbb{F}$ with $+$ as the additive operation and $\cdot$ as the multiplicative operation, a commutative group $(V, +)$ with an operation of $\mathbb{F}$ onto V is a *vector space over* $\mathbb{F}$ if for all $a, b \in \mathbb{F}$ and $x, y \in V$, we have: **i)** $a(bx) = (ab)x$; **ii)** $a(x + y) = ax + ay$; **iii)** $(a + b)x = ax + bx$; **iv)** $1 \cdot x = x$. Here, 1 is the additive identity of the field $\mathbb{F}$. If V is a vector space over $\mathbb{F}$, then the elements of V are called vectors. One of the natural candidates for vector spaces over a field $\mathbb{F}$ is $\mathbb{F}^n$, where $n \in \mathbb{N}$ and the function $f(\cdot)$ is the component-wise multiplication. In this paper, we restrict ourselves only to such types of vector spaces.

In the following, consider a field $\mathbb{F}$ and a vector space $V = \mathbb{F}^n$, where $n \in \mathbb{N}$. For a vector $\mathbf{v} = (b_1, b_2, \ldots, b_n) \in \mathbb{F}^n$ and an integer $i \in [n]$, by $\mathbf{v}[i]$ we denote the ith element (or entry) of $\mathbf{v}$, i.e., the element b_i. For vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t \in \mathbb{F}^n$, a linear combination of them is a vector $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_t\mathbf{v}_t$, where $a_1, a_2, \ldots, a_t \in \mathbb{F}$. Furthermore, a *linear relation* among them is exhibited when $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_t\mathbf{v}_t = 0$, for some $a_1, a_2, \ldots, a_t \in \mathbb{F}$. In the above, the a_is are called the *coefficients*. A set of vectors is said to be *linearly independent* if there is no linear relation among them except the trivial one, where each of the coefficients is 0. A set of vectors that is not linearly independent is said to be *linearly dependent*. An inclusion-wise maximal set of linearly independent vectors is called a *basis* of the vector space. It is known that for bases B, B' of a vector space, we have $|B| = |B'|$. By $\mathbb{F}_2$ we denote the field with exactly two elements, namely 0 and 1, with the usual addition and multiplication modulo 2 as the field operations. For two vectors $\mathbf{u}, \mathbf{v} \in V'$, $\mathbf{u} \cdot \mathbf{v}$ denotes the dot product of these two vectors. We refer the reader to [41] for more details on linear algebra.

3 Computing a Redundant Solution

Let (G, k) be an instance of IVD. A subset $S \subseteq V(G)$ such that $G - S$ is an interval graph is called a *solution*, and a solution of size at most t is called a *t-solution*. Towards the definition of redundancy, we need to introduce a few simple notions related to hitting and covering. Given a family $\mathcal{W} \subseteq 2^{V(G)}$, we say that a subset $S \subseteq V(G)$ *hits* $\mathcal{W}$ if for all $W \in \mathcal{W}$, we have $S \cap W \neq \emptyset$. A family $\mathcal{W} \subseteq 2^{V(G)}$ is *t-necessary* if every solution of size at most t hits $\mathcal{W}$. Moreover, we say that an obstruction $\mathbb{O}$ is *covered by* $\mathcal{W}$ if there exists $W \in \mathcal{W}$, such that $W \subseteq V(\mathbb{O})$. Now, we are ready to formally define our notion of redundancy.

Definition 3.1. Given a family $\mathcal{W} \subseteq 2^{V(G)}$ and $t \in \mathbb{N}$, a subset $M \subseteq V(G)$ is *t-redundant with respect to* $\mathcal{W}$ if for every obstruction $\mathbb{O}$ that is not covered by $\mathcal{W}$, it holds that $|M \cap V(\mathbb{O})| > t$.

The purpose of this section is to prove Lemma 3.1 below. Intuitively, this lemma asserts that an r-redundant solution M whose size is polynomial in k (for a fixed constant r) can be computed in polynomial time. Such a set M plays a central role in all of our subsequent reduction rules that comprise our kernelization algorithm. We remark that in this statement we use the letter ℓ rather than k to avoid confusion, as we will use this result with $\ell = k + 2$.

Lemma 3.1. *Let $r \in \mathbb{N}$ be a fixed constant, and (G, ℓ) be an instance of* IVD*. In polynomial time, it is possible to either conclude that (G, ℓ) is a* No*-instance, or compute an ℓ-necessary family $\mathcal{W} \subseteq 2^{V(G)}$ and a set $M \subseteq V(G)$, such that $\mathcal{W} \subseteq 2^M$ and M is a $(r+1)(6\ell)^{r+1}$-solution that is r-redundant with respect to $\mathcal{W}$.*

A central component in our proof of Lemma 3.1 is an approximation algorithm for IVD, given by Cao [9]:

Proposition 3.1. ([9]) IVD *admits a polynomial-time 6-approximation algorithm, called* `ApproxIVD`*.*

In particular, a main idea in our proof is to iteratively grow the redundancy of a solution by making calls to this approximation algorithm. Besides Proposition 3.1, towards the proof of Lemma 3.1, we give a simple definition of a graph on which we will apply the approximation algorithm and hence determine whether a set of vertices should be added to $\mathcal{W}$.

Definition 3.2. Let G be a graph, $U \subseteq V(G)$, and $t \in \mathbb{N}$. Then, $\mathsf{copy}(G, U, t)$ is defined as the graph G' on the vertex set $V(G) \cup \{v^i \mid v \in U, i \in [t]\}$ and the edge set $E(G) \cup \{(u^i, v) \mid (u, v) \in E(G), u \in U, i \in [t]\} \cup \{(u^i, v^j) \mid (u, v) \in E(G), u, v \in U, i, j \in [t]\} \cup \{(v, v^i) \mid v \in U, i \in [t]\} \cup \{(v^i, v^j) \mid v \in U, i, j \in [t], i \neq j\}$.

Informally, $\mathsf{copy}(G, U, t)$ is simply the graph G where for every vertex $u \in U$, we add t twins that (together with u) form a clique. Intuitively, this operation allows us to make a vertex set "undeletable"; in particular, this enables us to test later whether a vertex set is "redundant" and hence we can grow the redundancy of our solution, or whether it is "necessary" and hence we should update $\mathcal{W}$ accordingly. Before we turn to discuss computational issues, let us first assert that the operation in Definition 3.2 does not makes an interval graph become a non-interval graph. This is a basic requirement to verify before turning to design the above mentioned test.

Lemma 3.2. *Let G be a graph, $U \subseteq V(G)$, and $t \in \mathbb{N}$. If G is an interval graph, then $G' = \mathsf{copy}(G, U, t)$ is an interval graph as well.*

Proof. Suppose that G is an interval graph. Then, by Proposition 2.1, G admits a clique path (P, β). Now, we define (P', β') as follows: $P' = P$, and for all $x \in V(P')$, $\beta'(x) = \beta(x) \cup \{v^i \mid v \in \beta(x) \cap U, i \in [t]\}$. We claim that (P', β') is a clique path for G'. By using the fact that (P, β) is a path decomposition of G, we directly have the following properties. First, it is clear that $\bigcup_{x \in V(P')} \beta'(x) = V(G')$. Second, for any edge $e = (u, v) \in E(G')$ such that $u, v \in V(G)$, there exists $x_e \in V(P')$ such that $u, v \in \beta'(x_e)$. Then, since for all $v \in U$ and $i \in [t]$, it holds that $\beta'^{-1}(v) = \beta'^{-1}(v^i)$, we derive that for any edge $(u', v') \in E(G')$ there is a node $x \in V(P')$ such that $u', v' \in \beta'(x)$. Third, for any $v \in V(G)$, the collection of nodes $P'_v = \{x \in V(P') \mid v \in \beta'(u)\}$ is a subpath of P', and since for any $v \in U$ and $i \in [t]$, it holds that $\beta'^{-1}(v) = \beta'^{-1}(v^i)$, we derive that for any $v' \in V(G')$, the collection of nodes $P'_{v'} = \{x \in V(P') \mid v' \in \beta'(x)\}$ is a subpath of P'. Now, note that for all $x \in V(P')$, $\beta(x)$ is a clique, and for all $u, v \in \beta(x)$ (possibly $u = v$) and $i, j \in [t]$, u^i is adjacent to u, u^j (if $i \neq j$), v and v^j, which implies that $\beta'(x)$ is also a clique. Hence, (P', β') is indeed clique path for G'. By Proposition 2.1, we derive that G' is an interval graph. $\square$

Now, let us present two simple claims that exhibit relations between the algorithm `ApproxIVD` and Definition 3.2. After presenting these two claims, we will be ready to give our algorithm for computing a redundant solution. Roughly speaking, the first claim exhibits the meaning of a situation where ApproxIVD returns a "large" solution; intuitively, for the purpose of the design of our algorithm, we interpret this meaning as an indicator to extend $\mathcal{W}$.

LEMMA 3.3. *Let G be a graph, $U \subseteq V(G)$, and $\ell \in \mathbb{N}$. If the algorithm ApproxIVD returns a set A of size larger than 6ℓ when called with $G' = \mathsf{copy}(G, U, 6\ell+1)$ as input, then $\{U\}$ is ℓ-necessary.*

Proof. Suppose that ApproxIVD returns a set A of size larger than 6ℓ when called with G' as input. Then, (G', ℓ) is a No-instance. Suppose, by way of contradiction, that $\{U\}$ is not ℓ-necessary. Then, G has an ℓ-solution S such that $S \cap U = \emptyset$. In particular, $\widehat{G} = G - S$ is an interval graph such that $U \subseteq V(\widehat{G})$. However, this means that $\mathsf{copy}(\widehat{G}, U, 6\ell+1) = G' - S$, which by Lemma 3.2 implies that $G' - S$ is an interval graph. Thus, S is an ℓ-solution for G', which is a contradiction (as (G', ℓ) is a No-instance). ☐

Complementing our first claim, the second claim exhibits the meaning of a situation where ApproxIVD returns a "small" solution A; we interpret this meaning as an indicator to grow the redundancy of our current solution M by adding A—indeed, this lemma implies that every obstruction is hit one more time when adding A to a subset $U \subseteq M$ (to grow the redundancy of M, every subset $U \subseteq M$ will have to be considered).

LEMMA 3.4. *Let G be a graph, $U \subseteq V(G)$, and $\ell \in \mathbb{N}$. If the algorithm ApproxIVD returns a set A of size at most 6ℓ when called with $G' = \mathsf{copy}(G, U, 6\ell+1)$ as input, then for every obstruction $\mathbb{O}$ of G, $|V(\mathbb{O}) \cap U| + 1 \leq |V(\mathbb{O}) \cap (U \cup (A \cap V(G)))|$.*

Proof. Suppose that ApproxIVD returned a set A of size at most 6ℓ when called with G' as input. Let $\mathbb{O}$ be some obstruction of G, and denote $B = V(\mathbb{O}) \cap U$. Since $|A| \leq 6\ell$, for every vertex $v \in B$, we have that $v \in V(G') \setminus A$ or there exists $i(v) = i \in [6\ell]$ such that $v^i \in V(G') \setminus A$. Moreover, we have that the graph obtained from $\mathbb{O}$ by replacing each vertex $v \in B \cap A$ by $v^{i(v)}$ is an obstruction (as v and $v^{i(v)}$ are twins). Thus, as A is a solution for G', there exists $v \in V(G) \setminus B$ such that $v \in A \cap V(\mathbb{O})$. Hence, we have that $|V(\mathbb{O}) \cap U| + 1 \leq |V(\mathbb{O}) \cap (U \cup (A \cap V(G)))|$. ☐

Now, let us describe our algorithm, RedundantIVD, to compute a redundant solution. First, RedundantIVD initializes M_0 to be the output obtained by calling the algorithm ApproxIVD with G as input, $\mathcal{W}_0 := \emptyset$ and $\mathcal{T}_0 := \{(v) \mid v \in M_0\}$. If $|M_0| > 6\ell$, then RedundantIVD concludes that (G, ℓ) is a No-instance. Otherwise, for $i = 1, 2, \ldots, r$ (in this order), the algorithm executes the following steps:

1. Initialize $M_i := M_{i-1}$, $\mathcal{W}_i := \mathcal{W}_{i-1}$ and $\mathcal{T}_i := \emptyset$.
2. For every tuple $(v_0, v_1, \ldots, v_{i-1}) \in \mathcal{T}_{i-1}$:
   (a) Let A be the output obtained by calling the algorithm ApproxIVD with $\mathsf{copy}(G, \{v_0, v_1, \ldots, v_{i-1}\}, 6\ell+1)$ as input.
   (b) If $|A| > 6\ell$, then insert $\{v_0, v_1, \ldots, v_{i-1}\}$ into $\mathcal{W}_i$.
   (c) Otherwise, insert every vertex in $(A \cap V(G)) \setminus \{v_0, v_1, \ldots, v_{i-1}\}$ into M_i, and for all $u \in (A \cap V(G)) \setminus \{v_0, v_1, \ldots, v_{i-1}\}$, insert $(v_0, v_1, \ldots, v_{i-1}, u)$ into $\mathcal{T}_i$.

Eventually, the algorithm outputs the pair $(M_r, \mathcal{W}_r)$.

Let us comment that in this algorithm, we make use of the sets $\mathcal{T}_{i-1}$ rather than going over all subsets of size i of M_{i-1} in order to obtain a substantially better algorithm in terms of the size of the produced redundant solution.

The properties of the algorithm RedundantIVD that are relevant to us are summarized in the following lemma and observation, which are proved by induction and by making use of Lemmata 3.2, 3.3 and 3.4. Roughly speaking, we first assert that, unless (G, ℓ) is concluded to be a No-instance, we compute sets $\mathcal{W}_i$ that are ℓ-necessary as well as that the tuples in $\mathcal{T}_i$ "hit more vertices" of the obstructions in the input as i grows larger.

LEMMA 3.5. *Consider a call to RedundantIVD with (G, ℓ, r) as input that did not conclude that (G, ℓ) is a No-instance. For all $i \in [r]_0$, the following conditions hold:*

1. *For any set $W \in \mathcal{W}_i$, every solution S of size at most ℓ satisfies $W \cap S \neq \emptyset$.*
2. *For any obstruction $\mathbb{O}$ of G that is not covered by $\mathcal{W}_i$, there exists $(v_0, v_1, \ldots, v_i) \in \mathcal{T}_i$ such that $\{v_0, v_1, \ldots, v_i\} \subseteq V(\mathbb{O})$.*

Proof. The proof is by induction on i. In the base case, where $i = 0$, Condition 1 trivially holds as $\mathcal{W}_0 = \emptyset$, and Condition 2 holds as M_0 is a solution and $\mathcal{T}_0$ simply contains a 1-vertex tuple for every vertex in M_0. Now, suppose that the claim is true for $i - 1 \geq 0$, and let us prove it for i.

To prove Condition 1, consider some set $W \in \mathcal{W}_i$. If $W \in \mathcal{W}_{i-1}$, then by the inductive hypothesis, every solution of size at most ℓ satisfies $W \cap S \neq \emptyset$. Thus, we next suppose that $W \in \mathcal{W}_i \setminus \mathcal{W}_{i-1}$. Then, there exists a tuple $(v_0, v_1, \ldots, v_{i-1}) \in \mathcal{T}_{i-1}$ in whose iteration RedundantIVD inserted $W = \{v_0, v_1, \ldots, v_{i-1}\}$ into $\mathcal{W}_i$. In that iteration, ApproxIVD was called with

$\mathsf{copy}(G, W, 6\ell+1)$ as input, and returned a set A of size larger than 6ℓ. Thus, by Lemma 3.3, every solution S of size at most ℓ satisfies $W \cap S \neq \emptyset$.

To prove Condition 2, consider some obstruction $\mathbb{O}$ of G that is not covered by $\mathcal{W}_i$. By the inductive hypothesis and since $\mathcal{W}_{i-1} \subseteq \mathcal{W}_i$, there exists a tuple $(v_0, v_1, \ldots, v_{i-1}) \in \mathcal{T}_{i-1}$ such that $\{v_0, v_1, \ldots, v_{i-1}\} \subseteq V(\mathbb{O})$. Consider the iteration of `RedundantIVD` corresponding to this tuple, and denote $U = \{v_0, v_1, \ldots, v_{i-1}\}$. In that iteration, `ApproxIVD` was called with $\mathsf{copy}(G, U, 6\ell+1)$ as input, and returned a set A of size at most 6ℓ. By Lemma 3.4, $|V(\mathbb{O}) \cap U| + 1 \leq |V(\mathbb{O}) \cap (U \cup (A \cap V(G)))|$. Thus, there exists $v_i \in (A \cap V(G)) \setminus U$ such that $U \cup \{v_i\} \subseteq V(\mathbb{O})$. However, by the specification of `ApproxIVD`, this means that there exists $(v_0, v_1, \ldots, v_i) \in \mathcal{T}_i$ such that $\{v_0, v_1, \ldots, v_i\} \subseteq V(\mathbb{O})$. $\square$

Towards showing that the output set M_r is "small", let us upper bound the sizes of the sets M_i and $\mathcal{T}_i$.

OBSERVATION 3.1. *Consider a call to `RedundantIVD` with (G, ℓ, r) as input that did not conclude that (G, ℓ) is a No-instance. For all $i \in [r]_0$, $|M_i| \leq \sum_{j=0}^{i} (6\ell)^{j+1}$, $|\mathcal{T}_i| \leq (6\ell)^{i+1}$ and every tuple in $\mathcal{T}_i$ consists of distinct vertices.*

Proof. The proof is by induction on i. In the base case, where $i = 0$, the correctness follows as `ApproxIVD` returned a set of size at most 6ℓ. Now, suppose that the claim is true for $i-1 \geq 0$, and let us prove it for i. By the specification of the algorithm and inductive hypothesis, we have that $|M_i| \leq |M_{i-1}| + 6\ell|\mathcal{T}_{i-1}| \leq \sum_{j=1}^{i+1} (6\ell)^j$ and $|\mathcal{T}_i| \leq 6\ell|\mathcal{T}_{i-1}| \leq (6\ell)^{i+1}$. Moreover, by the inductive hypothesis, for every tuple in $\mathcal{T}_i$, the first i vertices are distinct, and by the specification of `ApproxIVD`, the last vertex is not equal to any of them. $\square$

By the specification of `RedundantIVD`, as a corollary to Lemma 3.5 and Observation 3.1, we directly obtain the following result.

COROLLARY 3.1. *Consider a call to `RedundantIVD` with (G, ℓ, r) as input that did not conclude that (G, ℓ) is a No-instance. For all $i \in [r]_0$, $\mathcal{W}_i$ is an ℓ-necessary and M_i is a $\sum_{j=0}^{i} (6\ell)^{j+1}$-solution that is i-redundant with respect to $\mathcal{W}_i$.*

Clearly, `RedundantIVD` runs in polynomial time (as r is a fixed constant), and by the correctness of `ApproxIVD`, if it concludes that (G, ℓ) is a No-instance, then this decision is correct. Thus, since $\sum_{i=0}^{r} (6\ell)^{r+1} \leq (r+1)(6\ell)^{r+1}$, the correctness of Lemma 3.1 now directly follows as a special case of Corollary 3.1. Thus, our proof of Lemma 3.1 is complete.

In light of Lemma 3.1, from now on, we suppose that we have a $(k+2)$-necessary family $\mathcal{W} \subseteq 2^{V(G)}$ along with a $(r+1)(6(k+2))^{r+1}$-solution M that is r-redundant with respect to $\mathcal{W}$ for $r = 9$. Let us note that, any obstruction in G that is not covered by $\mathcal{W}$ intersects M in at least ten vertices. We have the following reduction rule that follows immediately from Lemma 3.5.

REDUCTION RULE 3.1. *Let v be a vertex such that $\{v\} \in \mathcal{W}$. Then, output the instance $(G - \{v\}, k-1)$.*

Henceforward, we will assume that each set in $\mathcal{W}$ has size at least 2.

4 Handling Module Components

Let (G, k) be an instance of IVD. Let us explicitly recap the steps taken so far, and then state our current objective in this context. First, we call Lemma 3.1 with $r = 9$ and $\ell = k+2$, and one of the following holds. If (in polynomial time) we conclude that $(G, k+2)$ is a No-instance, then we can (correctly) conclude that (G, k) is a No-instance as well. Otherwise, in polynomial time we obtain a $(k+2)$-necessary family $\mathcal{W} \subseteq 2^{V(G)}$ and a set $M \subseteq V(G)$, such that $\mathcal{W} \subseteq 2^M$ and M is a $10(6(k+2))^{10}$-solution that is 9-redundant with respect to $\mathcal{W}$. Furthermore, each set in $\mathcal{W}$ has size at least 2. The main goal of this section is to bound the total number of vertices across all module connected components of $G - M$. We remark that we will prove a slightly more general result, as it will be used later in our algorithm. Before that, we provide a simple reduction rule to bound the number of non-module components.

Bounding the Number of Non-Module Components. Let $\mathcal{C}$ denote the set of connected components of $G - M$. Moreover, we let $\mathcal{D}$ denote the set of connected components in $\mathcal{C}$ that are modules, and $\overline{\mathcal{D}} = \mathcal{C} \setminus \mathcal{D}$. To bound the size of $\overline{\mathcal{D}}$, we apply the following reduction rule.

REDUCTION RULE 4.1. *Suppose that there exist $v \in M$ and a set $\mathcal{A} \subseteq \overline{\mathcal{D}}$ of size $k+3$ such that for each $D \in \mathcal{A}$, there exist $u, w \in V(D)$ such that $u \in N_G(v)$ and $w \notin N_G(v)$. Then, output the instance $(G - \{v\}, k-1)$.*

LEMMA 4.1. *Reduction Rule 4.1 is safe.*

Proof. In one direction, suppose that (G, k) is a Yes-instance, and let S be a k-solution for G. Since $|\mathcal{A}| \geq k+3$, there exist three connected components $D_1, D_2, D_2 \in \overline{\mathcal{D}} \cap \mathcal{A}$ such that $S \cap (V(D_1) \cup V(D_2) \cup V(D_3)) = \emptyset$. However, for each $i \in [3]$, the subgraph of G induced by the vertex set consisting of v, together with an edge e in D_i with one endpoint of e being a neighbor of v and the other endpoint of e being a non-neighbor of v, is a long claw. Here, we relied on the fact that for each $i \in [3]$, D_i is connected. Thus, as $G - S$ is an interval graph, we derive that $v \in S$, and therefore $S \setminus \{v\}$ is a $(k-1)$-solution for $G - \{v\}$.

In the other direction, it is clear that if $(G-\{v\}, k-1)$ is a Yes-instance, then (G, k) is a Yes-instance. $\square$

We now observe that our rule indeed bounds the size of $\overline{\mathcal{D}}$.

OBSERVATION 4.1. *After the exhaustive application of Reduction Rule 4.1,* $|\overline{\mathcal{D}}| \leq (k+2)|M|$.

Proof. After the exhaustive application of Reduction Rule 4.1, every vertex in M has at most $k+2$ connected components in $\mathcal{C}$ where it has both a neighbor and a non-neighbor. Since for a connected component in $\overline{\mathcal{D}}$ that is not a module, there must exist a vertex in M that has both a neighbor and a non-neighbor in that component, we conclude that the observation is correct. $\square$

The Main Lemma of this Section. From now on, we focus on the main goal of this section: bound the total number of vertices in $\mathcal{D}$. As mentioned earlier, the arguments used to derive this bound will also be necessary at a later stage of our kernelization algorithm, and hence we present our goal in the form of a more general statement:

LEMMA 4.2. *Let* $\widehat{M} \subseteq V(G)$*, and* $\widehat{\mathcal{C}}$ *be some set of connected components of* $G-(M\cup\widehat{M})$ *that are modules. In polynomial time, it is possible to either output an instance* (G', k) *equivalent to* (G, k) *where* G' *is a strict (induced) subgraph of* G*, or to compute a subset* $B \subseteq V(\widehat{\mathcal{C}})$ *of size at most* $4(k+1)^2|M \cup \widehat{M}|^6$*, such that for any subset* $S \subseteq V(G)$ *of size at most* k*, the following property holds: If there exists an obstruction* $\mathbb{O}$ *for* G *that is not covered by* $\mathcal{W}$ *and such that* $V(\mathbb{O}) \cap S = \emptyset$*, there exists an obstruction* $\mathbb{O}'$ *for* G *such that* $V(\mathbb{O}') \cap S = \emptyset$ *and* $V(\mathbb{O}') \cap (V(\widehat{\mathcal{C}}) \setminus B) = \emptyset$.

Intuitively, the statement of this lemma expands M to $M \cup \widehat{M}$, and zooms into a subset $\widehat{\mathcal{C}}$ of the set of connected components that are modules in $G-(M\cup\widehat{M})$. Then, either it enables us to reduce the instance, or it produces a "small" subset $B \subseteq V(\widehat{\mathcal{C}})$ and implies that we need not "worry" about obstructions that intersect $V(\widehat{\mathcal{C}}) \setminus B$ but not B—if such an obstruction is not hit, then there is an obstruction that does not intersect $V(\widehat{\mathcal{C}}) \setminus B$ and which is not hit as well.

Let us now show that having Lemma 4.2 at hand, we can indeed bound the total number of vertices in all module components.

REDUCTION RULE 4.2. *Let* X *be the output of the algorithm in Lemma 4.2 when called with* $\widehat{M} = \emptyset$ *and* $\widehat{\mathcal{C}} = \mathcal{D}$*. If* X *is an instance* (G', k)*, then output* X*. Otherwise,* X *is a set* $B \subseteq V(\mathcal{D})$*, and we output the instance* $(G - \{v\}, k)$ *for a vertex* v *arbitrarily chosen from* $V(\mathcal{D}) \setminus B$.

By using Lemma 4.2, we derive the safeness of Reduction Rule 4.2.

LEMMA 4.3. *Reduction Rule 4.2 is safe.*

Proof. If X is an instance (G', k), then Lemma 4.2 directly implies that the rule is safe. Thus, we next suppose that $X = B$. In one direction, it is clear that if (G, k) is a Yes-instance, then $(G - \{v\}, k)$ is a Yes-instance as well.

In the other direction, suppose that $(G - \{v\}, k)$ is a Yes-instance. Let S be a k-solution for $G - \{v\}$. We claim that S is also a k-solution for G. Suppose, by way of contradiction, that this claim is false. Then, there exists an obstruction $\mathbb{O}$ for $G - S$. As $S \cup \{v\}$ is a $(k+1)$-solution for G and $\mathcal{W}$ is $(k+2)$-necessary, we have that $S \cup \{v\}$ hits $\mathcal{W}$. Since $v \notin M$ and $\mathcal{W} \subseteq 2^M$, we derive that S hits $\mathcal{W}$. Thus, since $\mathbb{O}$ is an obstruction for $G-S$, we deduce that $\mathbb{O}$ is not covered by $\mathcal{W}$. Hence, by Lemma 4.2, there exists an obstruction $\mathbb{O}'$ for G such that $V(\mathbb{O}') \cap S = \emptyset$ and $V(\mathbb{O}') \cap (V(\mathcal{D}) \setminus B) = \emptyset$. However, as $v \in V(\mathcal{D}) \setminus B$, this implies that $\mathbb{O}'$ is also an obstruction for $(G - \{v\}) - S$, which is a contradiction as S is a k-solution for $G - \{v\}$. $\square$

Due to Reduction Rule 4.2, we have the following result.

OBSERVATION 4.2. *After the exhaustive application of Reduction Rule 4.2,* $|V(\mathcal{D})| \leq 4(k+1)^2|M|^6$.

We now turn to prove Lemma 4.2. In what follows, $\widehat{M}$ and $\widehat{\mathcal{C}}$ are as stated in this lemma. We denote $M' = M \cup \widehat{M}$. Note that since M is 9-redundant with respect to $\mathcal{W}$, we have that M' is also 9-redundant with respect to $\mathcal{W}$. We begin our proof by showing that the common neighborhood outside M' of any two non-adjacent vertices, unless these two vertices form a pair in $\mathcal{W}$, is simply a clique. This simple claim will come in handy in several arguments later.

LEMMA 4.4. *Let* $u, v \in V(G)$ *be distinct vertices such that* $(u, v) \notin E(G)$ *and* $\{u, v\} \notin \mathcal{W}$*. Then,* $G[(N_G(u) \cap N_G(v)) \setminus M']$ *is a clique.*

Proof. Suppose, by way of contradiction, that $G[N_G(u) \cap N_G(v) \setminus M']$ is not a clique. Then, there exist two vertices $x, y \in (N_G(u) \cap N_G(v)) \setminus M'$ that are not neighbors in G. Note that $\mathbb{O} = G[\{u, v, x, y\}]$ is a hole, and that $M \cap V(\mathbb{O}) \subseteq \{u, v\}$. Moreover, $\mathbb{O}$ is not covered by $\mathcal{W}$ (because $\{u, v\} \notin \mathcal{W}$ and every set in $\mathcal{W}$ has size at least 2). Since M is 9-redundant, this means that $|M \cap V(\mathbb{O})| > 9$. However, $|V(\mathbb{O})| = 4$, hence we have reached a contradiction. $\square$

Structure of Obstructions Intersecting Module Components. In order to reduce our instance or to obtain a set B as required to prove Lemma 4.2, we need to understand how obstructions can intersect module components. For this purpose, we state a simple proposition by Cao and Marx [10]. This proposition asserts that because we are dealing with modules, these intersections are quite restricted.

PROPOSITION 4.1. ([10]) *Let C be a module in G and $\mathbb{O}$ be a minimal obstruction. If $|V(\mathbb{O})| > 4$, then either $V(\mathbb{O}) \subseteq V(C)$ or $|V(\mathbb{O}) \cap V(C)| \leq 1$.*

By Proposition 4.1, we directly obtain the following lemma.

LEMMA 4.5. *Let C be a module such that $V(C) \cap M' = \emptyset$, and let $\mathbb{O}$ be a minimal obstruction that is not covered by $\mathcal{W}$. Then, $|V(\mathbb{O}) \cap V(C)| \leq 1$.*

Proof. Since $\mathbb{O}$ is an obstruction that is not covered by $\mathcal{W}$, it holds that $|M' \cap V(\mathbb{O})| > 9$. In particular, as $V(C) \cap M' = \emptyset$, we have that $|V(\mathbb{O})| > 4$ and $V(\mathbb{O}) \setminus V(C) \neq \emptyset$. Then, as C is a module and $\mathbb{O}$ is minimal, by Proposition 4.1, we have that $|V(\mathbb{O}) \cap V(C)| \leq 1$. $\square$

Reducing the Size of Module Components. To ensure we have only small module components, we apply the following rule.

REDUCTION RULE 4.3. *Suppose that there exists $C \in \widehat{\mathcal{C}}$ such that $|V(C)| > k + 1$. Then, output the instance $(G - \{v\}, k)$, where v is an arbitrarily chosen vertex of C.*

LEMMA 4.6. *Reduction Rule 4.3 is safe.*

Proof. In one direction, it is clear that if (G, k) is a Yes-instance, then $(G - \{v\}, k)$ is a Yes-instance as well.

In the other direction, suppose that $(G - \{v\}, k)$ is a Yes-instance. Let S be a k-solution for $G - \{v\}$. We claim that S is also a k-solution for G. Suppose, by way of contradiction, that this claim is false. Then, there exists a minimal obstruction $\mathbb{O}$ for $G - S$. As $S \cup \{v\}$ is a $(k+1)$-solution for G and $\mathcal{W}$ is $(k+2)$-necessary, we have that $S \cup \{v\}$ hits $\mathcal{W}$. Since $v \notin M$ and $\mathcal{W} \subseteq 2^M$, we derive that S hits $\mathcal{W}$. Thus, since $\mathbb{O}$ is an obstruction for $G - S$, we deduce that $\mathbb{O}$ is not covered by $\mathcal{W}$. Hence, by Lemma 4.5, $|V(\mathbb{O}) \cap V(C)| \leq 1$. Thus, $V(\mathbb{O}) \cap V(C) = \{v\}$. Then, as C is a module, for any vertex $u \in V(C)$, it holds that $G[(V(\mathbb{O}) \setminus \{v\}) \cup \{u\}]$ is an obstruction. Since $|V(C)| > k + 1$, we have that $V(C) \setminus (S \cup \{v\}) \neq \emptyset$. However, this implies that there exists an obstruction $\mathbb{O}'$ for $(G - \{v\}) - S$, which is a contradiction as S is a k-solution for $G - \{v\}$. $\square$

Preliminary Marking Scheme. By Lemma 4.4, for all $u, v \in M'$ such that $(u, v) \notin E(G)$ and $\{u, v\} \notin \mathcal{W}$, there exists at most one $C \in \widehat{\mathcal{C}}$, denoted by C_{uv}, such that $N_G(u) \cap N_G(v) \cap V(C) \neq \emptyset$. Accordingly, denote

$$\mathcal{C}^\star = \{C_{uv} \in \widehat{\mathcal{C}} \mid u, v \in M', (u, v) \notin E(G), \{u, v\} \notin \mathcal{W}\}.$$

Moreover, denote $A^\star = V(\mathcal{C}^\star)$. From Reduction Rule 4.3, we have the following observation.

OBSERVATION 4.3. *The size of $A^\star$ is upper bounded by $(k + 1)|M'|^2$.*

Thus, in what follows, we do not need to "worry" about the modules in $\mathcal{C}^\star$ since we already know that they contain only few vertices in total. In the following, we proceed to analyze the modules in $\widehat{\mathcal{C}} \setminus \mathcal{C}^\star$. An important property of every vertex v in the modules in $\widehat{\mathcal{C}} \setminus \mathcal{C}^\star$, unlike the modules in $\mathcal{C}^\star$, is that every pair of vertices in its neighborhood in M' must be adjacent unless they form a set in $\mathcal{W}$.

OBSERVATION 4.4. *Consider a vertex $v \in V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$. For (distinct) vertices $u, w \in N_G(v) \cap M'$, at least one of $\{u, w\} \in \mathcal{W}$ or $(u, w) \in E(G)$ holds.*

Proof. For $v \in V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$, and (distinct) vertices $u, w \in N_G(v) \cap M'$, if one of $\{u, w\} \in \mathcal{W}$ or $(u, v) \in E(G)$ holds, then the claim trivially holds. Therefore, we assume that $\{u, w\} \notin \mathcal{W}$ and $(u, v) \notin E(G)$. Recall that each set in $\mathcal{W}$ is of size at least 2 (since Reduction Rule 3.1 is not applicable). From the above discussions, together with Lemma 4.4 we obtain that there is at most one connected component $C_{uw} \in \widehat{\mathcal{C}}$, such that $N_G(u) \cap N_G(w) \cap V(C_{uw}) \neq \emptyset$. Since $u, w \in N_G(v)$, it must be the case that $v \in C_{uw}$. But by our preliminary marking scheme, $C_{uw} \in \mathcal{C}^\star$. This contradicts that $v \in V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$. $\square$

Let us also consider the relation between obstructions and the modules in $\widehat{\mathcal{C}} \setminus \mathcal{C}^\star$. Roughly speaking, the following lemma already implies that we can focus on AWs of a very specific form. However, handling these obstructions requires a substantive amount of work in the rest of this section.

LEMMA 4.7. *Let $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$, and $\mathbb{O}$ be a minimal obstruction that is not covered by $\mathcal{W}$ such that $V(\mathbb{O}) \cap V(C) \neq \emptyset$. Then, $|V(\mathbb{O}) \cap V(C)| = 1$ and $\mathbb{O}$ is an AW where the vertex in $V(\mathbb{O}) \cap V(C)$ is a terminal.*

Proof. Consider $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$ and a minimal obstruction $\mathbb{O}$ that is not covered by $\mathcal{W}$, such that $V(\mathbb{O}) \cap V(C) \neq \emptyset$. First, as C is a module, from Lemma 4.5 we deduce that $|V(\mathbb{O}) \cap V(C)| = 1$. Furthermore, as $\mathbb{O}$ is not covered by $\mathcal{W}$, we have that $|V(\mathbb{O})| > 9$. This means that $\mathbb{O}$ is neither a long claw nor a whipping top. Let v be the unique vertex in $V(C) \cap V(\mathbb{O})$. If $\mathbb{O}$ is an induced cycle on at least 4 vertices, or one of the AWs where v is not one of the terminals, then $N_G(v) \cap V(\mathbb{O})$ contains a pair of non-adjacent vertices. But from Observation 4.4 together with the facts that $\mathbb{O}$ is not covered by $\mathcal{W}$ and $N_G(v) \subseteq V(C) \cup M$, for each $u, w \in N_G(v) \cap M' \cap V(\mathbb{O})$, we have $(u, v) \in E(G)$. Thus, we conclude that $\mathbb{O}$ is one of the AWs, where v is one of the terminals. ☐

Marking Scheme to Handle Non-Shallow Terminals. For every two subsets $X, Y \subseteq M'$ such that $|X| \leq 2$ and $|Y| \leq 2$, denote $A_{X,Y} = \{v \in V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star) \mid X \subseteq N_G(v), Y \cap N_G(v) = \emptyset\}$. Now, if $|A_{X,Y}| \leq k+1$, then define $A'_{X,Y} = A_{X,Y}$, and otherwise let $A'_{X,Y}$ be an arbitrarily chosen subset of size $k+1$ of $A_{X,Y}$. Let us denote $A' = \bigcup_{X,Y} A'_{X,Y}$, where X, Y range over all subsets $X, Y \subseteq M'$ such that $|X| \leq 2$ and $|Y| \leq 2$. Let us first observe that $|A'|$ is small.

OBSERVATION 4.5. *The size of A' is upper bounded by $(k+1)|M'|^4$.*

Now, let us verify that we have thus marked a set of vertices that is sufficient to “handle” non-shallow terminals. Roughly speaking, by this we mean that for any vertex v and obstruction $\mathbb{O}$ that satisfy the premise in this lemma, we can find $k+1$ “replacements” of v (so that we still have an obstruction) that belong to our marked set A'.

LEMMA 4.8. *Let $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$, $v \in V(C) \setminus A'$, and $\mathbb{O}$ be a minimal obstruction that is not covered by $\mathcal{W}$ such that $v \in V(\mathbb{O})$. If $\mathbb{O}$ is an AW where v is a non-shallow terminal, then there exists a subset $\hat{A} \subseteq A'$ of size $k+1$ such that for each $u \in \hat{A}$, $G[(V(\mathbb{O}) \setminus \{v\}) \cup \{u\}]$ contains an obstruction.*

Proof. First, by Lemma 4.7, we have that $\mathbb{O}$ is an AW such that $V(\mathbb{O}) \cap V(C) = \{v\}$ and v is a terminal of $\mathbb{O}$. Let us also note that $N_G(v) \subseteq M' \cup C$ and therefore $N_G(v) \cap V(\mathbb{O}) \subseteq M'$. Let $\mathbb{O}$ comprise of the base path $\mathsf{base}(\mathbb{O}) = (b_1, b_2, \ldots, b_z)$, non-shallow terminals t_ℓ and t_r, shallow terminal t, and centers c_1 and c_2 (as in the definition in Section 2). Here, if $\mathbb{O}$ is a †-AW, then we let $c = c_1 = c_2$. Suppose that v is not the shallow terminal of $\mathbb{O}$. Then, we have that v is either t_ℓ or t_r. Without loss of generality, suppose that $v = t_\ell$. Let us consider two cases, depending on whether $\mathbb{O}$ is a †-AW or a ‡-AW.

- Suppose that $\mathbb{O}$ is a †-AW. Notice that $b_1 \in M'$ as $(b_1, v) \in E(G)$, $V(\mathbb{O}) \cap V(C) = \{v\}$, and $N_G(v) \subseteq M' \cup C$. From Lemma 4.7 any vertex in $V(\mathbb{O}) \cap V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$ must be one of the terminals. Thus, we have $V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star) \cap (\{b_1, b_2, \ldots, b_z\} \cup \{c\}) = \emptyset$. We also recall that for each $u \in V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$, we have $N_G(u) \subseteq M' \cup V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$. In particular, if b_2 (or c) is not in M', no vertex in $V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$ can be adjacent to b_2 (or c). The above discussions together with the construction of A' implies the following: there exists a subset $Q \subseteq A'$ of $k+1$ vertices such that for each $u \in Q$, u is adjacent to b_1, and u is not adjacent to b_2 and c. Indeed, these are the vertices in the set $A'_{\{b_1\},\{b_2,c\} \cap M'}$ (the size of this set is $k+1$ since otherwise v should have belonged to it, but $v \notin A'$). Furthermore, b_1 is not adjacent to any vertex on $\mathbb{O}$ besides v, c and b_2. Therefore, for all $u \in Q$, using Observation 4.4 for obstructions not covered by $\mathcal{W}$, we have that u is not adjacent to any vertex on $V(\mathbb{O}) \cap M'$ besides b_1. Furthermore, for all $u \in Q$, since $N_G(u) \subseteq V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star) \cup M'$, we have that u is not adjacent to any vertex on $V(\mathbb{O}) \cap V(\mathcal{C}^\star)$. Lastly, because $V(\mathbb{O}) \cap V(C) = \{v\}$, for all $u \in Q$, we have that u is not adjacent to any vertex on $V(\mathbb{O}) \cap V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$ besides possibly v. Hence, for any vertex $u \in Q$, $G[(V(\mathbb{O}) \setminus \{v\}) \cup \{u\}]$ is also a †-AW.

- Suppose that $\mathbb{O}$ is a ‡-AW. Notice that $b_1, c_1 \in M'$ as $(b_1, v), (c_1, v) \in E(G)$, $V(\mathbb{O}) \cap V(C) = \{v\}$, and $N_G(v) \subseteq M' \cup C$. From Lemma 4.7 any vertex in $V(\mathbb{O}) \cap V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$ must be one of the terminals. Thus, we have $V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star) \cap (\{b_1, b_2, \ldots, b_z\} \cup \{c\}) = \emptyset$. We also recall that for each $u \in V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$, we have $N_G(u) \subseteq M' \cup V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star)$. The above discussions together with the construction of A' implies the following: there exists a subset $Q \subseteq A'$ of $k+1$ vertices $u \in A'$ such that u is adjacent to both c_1 and b_1, and u is adjacent to neither c_2 nor b_2. Indeed, these are the vertices in the set $A'_{\{b_1,c_1\},\{b_2,c_2\} \cap M'}$ (as in the previous case, the size of this set is $k+1$ since otherwise v should have belonged to it, but $v \notin A'$). Notice that b_1 is not adjacent to any vertex on $\mathbb{O}$ besides v, c_1, c_2 and b_2. For all $u \in Q$, using Observation 4.4 for obstructions not covered by $\mathcal{W}$ and the facts that

$N_G(u) \subseteq V(\widehat{\mathcal{C}} \setminus \mathcal{C}^\star) \cup M'$ and $V(\mathbb{O}) \cap V(C) = \{v\}$ (using the exact same rationale as in the previous case), we have that u is not adjacent to any vertex on $\mathbb{O} - \{v\}$ besides c_1 and b_1. Hence, for any vertex $u \in Q$, $G[(V(\mathbb{O}) \setminus \{v\}) \cup \{u\}]$ is also a ‡-AW.

In both cases, we derived the desired claim, and thus the proof is complete. □

Marking Scheme to Handle Shallow Terminals. For this part in our proof, we require the following notation: we say that a path P is *covered by* $\mathcal{W}$ if there is a set $W \in \mathcal{W}$ such that $W \subseteq V(P)$. Intuitively, we think of P as part of the base of an obstruction, hence the notation above is a natural extension of covering to this context.

Before we present our marking scheme, let us explicitly state the following observation, which follows from Observation 4.4 in the same manner as Lemma 4.7.

OBSERVATION 4.6. *Let P be an induced path in $G[V(G) \setminus V(C)]$ for some $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$ such that P is not covered by $\mathcal{W}$. For all $v \in V(C)$, $|N_G(v) \cap V(P)| \leq 2$, and if $|N_G(v) \cap V(P)| = 2$, then the two vertices in $N_G(v) \cap V(P)$ are adjacent on P.*

Proof. Consider $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$, $v \in V(C)$, and an induced path P in $G[V(G) \setminus V(C)]$ which is not covered by $\mathcal{W}$. If $|N_G(v) \cap V(P)| \leq 1$, then the claim trivially follows. Otherwise, we assume that $|N_G(v) \cap V(P)| \geq 2$. Consider (distinct) vertices $u, w \in N_G(v) \cap V(P)$. From Observation 4.4, we have that $(u, w) \in E(G)$. Here, we relied on the fact that P is not covered by $\mathcal{W}$. Since P is an induced path, u and w must be adjacent vertices in P. From the above we can conclude that v cannot have three neighbors in P as P is an induced path in G. Moreover, if v has two neighbors in P then they must be adjacent vertices. □

Denote $N = M' \cup A^\star \cup A'$. (Recall that $A^\star = V(\mathcal{C}^\star)$ and that A' is the set of vertices marked when we dealt with non-shallow terminals.) For all (not necessarily distinct) vertices $c_1, c_2 \in M'$, denote $A_{\{c_1,c_2\}} = \{v \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A') \mid \{c_1, c_2\} \subseteq N_G(v)\}$. Intuitively, $A_{\{c_1,c_2\}}$ is the set of vertices among the unmarked vertices in $\widehat{\mathcal{C}}$ that are neighbors of both c_1 and c_2 and hence can play the role of shallow terminals in obstructions having c_1 and c_2 as centers. Moreover, let us arbitrarily order N and $E(G[N])$ as follows: $N = \{v_1, v_2, \ldots, v_{|N|}\}$ and $E(G[N]) = \{e_1, e_2, \ldots, e_{|E(G[N])|}\}$. Thus, when we define vectors having $|N|$ or $|E(G[N])|$ entries below, we can work with a natural correspondence between the index of an entry in the vector and an element of N or $E(G[N])$, respectively.

In what follows, we begin the part in our analysis that is based on linear algebra. To this end, we first need to encode our problem in this language, which entails the introduction of appropriate notations. Afterwards, we will present a marking scheme based on these notations. The analysis of this scheme is done in a sequence of several lemmata, after which we will be ready to conclude the proof of Lemma 4.2.

First, with every vertex $u \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A')$, we associate two binary vectors that capture incidence relations between u and the elements (vertices and edges) in $G[N]$:

- **Vertex incidence relations.** $\mathsf{vinc}(u) = (b_1, b_2, \ldots, b_{|N|})$, where for all $i \in [|N|]$, $b_i = 1$ if and only if $v_i \in N_G(u)$;
- **Edge incidence relations.** $\mathsf{einc}(u) = (b_1, b_2, \ldots, b_{|E(G[N])|})$, where for all $i \in [|E(G[N])|]$, $b_i = 1$ if and only if u is adjacent to both endpoints of e_i.

Complete incidence relations. In addition, we define $\mathsf{inc}(u)$ as the vector that is the concatenation of $\mathsf{vinc}(u)$ and $\mathsf{einc}(u)$, to which we add 1 at the end. Formally, $\mathsf{inc}(u)$ is a binary vector with $|N| + |E(G[N])| + 1$ entries, where for all $i \in [|N|]$, the i^{th} entry of $\mathsf{inc}(u)$ equals the i^{th} entry of $\mathsf{vinc}(u)$, for all $i \in [|E(G[N])| + |N|] \setminus [|N|]$, the i^{th} entry of $\mathsf{inc}(u)$ equals the $(i - |N|)^{th}$ entry of $\mathsf{einc}(u)$, and the last entry of $\mathsf{inc}(u)$ is 1. These incidence vectors are associated with the vector space $\mathbb{F}_2^q$ for $q = |N| + |E(G[N])| + 1$, and all calculations related to these vectors are performed accordingly. This completes the description of the notations required to present our marking scheme.

For all (not necessarily distinct) vertices $c_1, c_2 \in M'$, we have the following subprocedure of our marking scheme. First, we define $\mathbf{V}_{\{c_1,c_2\}}$ to be the *multiset* $\{\mathsf{inc}(u) \mid u \in A_{\{c_1,c_2\}}\}$. More precisely, the number of occurrences of a vector in $\mathbf{V}_{\{c_1,c_2\}}$ equals the number of vertices $u \in A_{\{c_1,c_2\}}$ such that $\mathsf{inc}(u)$ equals that vector. Now, we proceed as follows.

1. Initialize $\widehat{\mathbf{V}}^0_{\{c_1,c_2\}} = \emptyset$.
2. For $i = 1, 2, \ldots, k+1$: compute some basis $\mathbf{B}^i_{\{c_1,c_2\}}$ for the vector subspace $\mathbf{V}_{\{c_1,c_2\}} \setminus \widehat{\mathbf{V}}^{i-1}_{\{c_1,c_2\}}$ (with respect to $\mathbb{F}_2^q$),[2] and denote $\widehat{\mathbf{V}}^i_{\{c_1,c_2\}} = \widehat{\mathbf{V}}^{i-1}_{\{c_1,c_2\}} \cup \mathbf{B}^i_{\{c_1,c_2\}}$.

[2]Here, note that the subtraction concerns multisets. In particular, if an element occurs x times in a multiset X, and y times in a multiset $Y \subseteq X$, then it occurs $x - y$ times in $X \setminus Y$.

3. For every occurrence of a vector $\mathbf{v} \in \widehat{\mathbf{V}}^{k+1}_{\{c_1,c_2\}}$, arbitrarily choose a unique vertex $u \in A_{\{c_1,c_2\}}$ such that $\mathsf{inc}(u) = \mathbf{v}$ and denote it by $u_{\mathbf{v}}$ (the existence of sufficiently many such distinct vertices directly follows from the definition of $\mathbf{V}_{\{c_1,c_2\}}$).

4. Denote $\widehat{A}_{\{c_1,c_2\}} = \{u_{\mathbf{v}} : \mathbf{v} \in \widehat{\mathbf{V}}^{k+1}_{\{c_1,c_2\}}\}$, and note that $\widehat{A}_{\{c_1,c_2\}}$ is a set (rather than a multiset).

Finally, having performed all subprocedures, we denote $\widehat{A} = \bigcup_{c_1,c_2 \in M'} \widehat{A}_{\{c_1,c_2\}}$. Here, union refers to sets, that is, every vertex occurs in $\widehat{A}$ once even if it belongs to more than one set of the form $\widehat{A}_{\{c_1,c_2\}}$. This completes the description of our marking scheme.

We proceed to analyze our marking scheme. Let us first observe that we have not marked "many" vertices, that is, we upper bound $|\widehat{A}|$. (Here, the bound $|N| \leq 2|M'|^4$ follows from Observations 4.3 and 4.5, and since $N = M' \cup A^\star \cup A'$.)

Lemma 4.9. *The size of $\widehat{A}$ is upper bounded by $(k+1)|M'|^2|N|^2 \leq 2(k+1)^2|M'|^6$.*

Proof. To show that $|\widehat{A}| \leq (k+1)|M'|^2|N|^2$, it is sufficient to show that for all $c_1, c_2 \in M'$, $|\widehat{A}_{\{c_1,c_2\}}| \leq (k+1)|N|^2$. To this end, consider some $c_1, c_2 \in M'$. Now, observe that the number of entries of the vectors in $\mathbf{V}_{\{c_1,c_2\}}$ is $q = |N|+|E(G[N])|+1 \leq |N|+\frac{|N|(|N|-1)}{2}+1 \leq |N|^2$ (assuming $|N| > 1$, as otherwise, we can obtain a trivial kernel). Hence, every basis of $\mathbf{V}_{\{c_1,c_2\}}$ (or of a subset of $\mathbf{V}_{\{c_1,c_2\}}$) is of size at most $|N|^2$. As $\widehat{\mathbf{V}}^{k+1}_{\{c_1,c_2\}}$ is a multiset that is the union of $(k+1)$ bases of $\mathbf{V}_{\{c_1,c_2\}}$ (or of subsets of $\mathbf{V}_{\{c_1,c_2\}}$), we have that $|\widehat{\mathbf{V}}^{k+1}_{\{c_1,c_2\}}| \leq (k+1)|N|^2$. Since $|\widehat{\mathbf{V}}^{k+1}_{\{c_1,c_2\}}| = |\widehat{A}_{\{c_1,c_2\}}|$, the proof is complete. $\square$

Now, let us verify that we have a set of vertices that is sufficient to "handle" shallow terminals. This will be done in a sequence of two lemmata and a corollary. For this purpose, we need the following notation where we alter incidence vectors by nullifying some of their entries.

- **Nullifying Subsets of Vertices and Edges.** Given a pair (X,Y), where $X \subseteq N$ and $Y \subseteq E(G[N])$, and a vertex $u \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A')$, we define $\mathsf{inc}^{X,Y}(u)$ to be the vector $\mathsf{inc}(u)$ where all entries associated with vertices and edges that do not belong to $X \cup Y$ are changed to 0. Formally, $\mathsf{inc}^{X,Y}(u)$ is a binary vector with $|N|+|E(G[N])|+1$ entries, where for all $i \in [|N|]$, the i^{th} entry of $\mathsf{inc}(u)$ equals the i^{th} entry of $\mathsf{vinc}(u)$ if $v_i \in X$ and to 0 otherwise, for all $i \in [|E(G[N])|+|N|] \setminus [|N|]$, the i^{th} entry of $\mathsf{inc}^{X,Y}(u)$ equals the $(i-|N|)^{th}$ entry of $\mathsf{einc}(u)$ if $e_{i-|N|} \in Y$ and to 0 otherwise, and the last entry of $\mathsf{inc}^{X,Y}(u)$ is 1.

- **Nullifying an Induced Path.** Furthermore, for an induced path P in $G - (V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A'))$ and a vertex $u \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A')$, we denote $\mathsf{inc}^P(u) = \mathsf{inc}^{X,Y}(u)$ where $X = V(P) \cap N$ and $Y = E(P) \cap E(G[N])$.

Moreover, recall that given a vector $\mathbf{v}$ and an entry index i, $\mathbf{v}[i]$ denotes the i^{th} entry of $\mathbf{v}$.

Lemma 4.10. *Let P be an induced path in $G[V(G) \setminus V(C)]$ for some $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$ such that P is not covered by $\mathcal{W}$. For all $u \in V(C)$, $\sum_{i=1}^{q} \mathsf{inc}^P(u)[i] = 1 \mod 2$ if and only if $N_G(u) \cap V(P) = \emptyset$.*

Proof. Consider some vertex $u \in V(C)$. For the reverse direction of the proof, suppose that $N_G(u) \cap V(P) = \emptyset$. Then, all of the entries of $\mathsf{inc}^P(u)$ equal 0, except for the last entry which equals 1. Thus, $\sum_{i=1}^{q} \mathsf{inc}^P(u)[i] = 1 \mod 2$.

For the forward direction of the proof, suppose that $N_G(u) \cap V(P) \neq \emptyset$. Then, by Observation 4.6, $|N_G(u) \cap V(P)|$ is either 1 or 2, and if it is 2, then the two vertices in $N_G(u) \cap V(P)$ are adjacent on P. Furthermore, observe that as $V(P) \cap V(C) = \emptyset$ and $N_G(u) \subseteq V(C) \cup M'$, we have that $N_G(u) \cap V(P) \subseteq M'$. Thus, in case $|N_G(u) \cap V(P)| = 1$, it follows that there exists exactly one entry in $\mathsf{inc}^P(u)$ that equals 1 apart from the last entry, which is the entry corresponding to the vertex in $N_G(u) \cap V(P)$. Moreover, in case $|N_G(u) \cap V(P)| = 2$, it follows that there exist exactly three entries in $\mathsf{inc}^P(u)$ that equal 1 apart from the last entry, which are the two entries corresponding to the two vertices in $N_G(u) \cap V(P)$ and the entry corresponding to the edge between these two vertices. In both cases, we derive that $\sum_{i=1}^{q} \mathsf{inc}^P(u)[i] = 0 \mod 2$ as desired. $\square$

The reason why we need Lemma 4.10 is that we make use of it in the proof of the following lemma. Informally, this lemma exhibits the existence of $k+1$ "replacements" for each unmarked shallow terminal.

Lemma 4.11. *Let $w \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A})$, and $\mathbb{O}$ be an AW that is not covered by $\mathcal{W}$ such that $V(\mathbb{O}) \cap (V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A})) = \{w\}$ and w is the shallow terminal of $\mathbb{O}$. Let $\{c_1, c_2\}$ be the set of centers of $\mathbb{O}$ (with $c_1 = c_2$ if $\mathbb{O}$ is a †-AW). Then, for all $i \in [k+1]$, there exists $\mathbf{v} \in \mathbf{B}^i_{\{c_1,c_2\}}$ such that $G[(V(\mathbb{O}) \setminus \{w\}) \cup \{u_{\mathbf{v}}\}]$ is an obstruction.*

Proof. Consider some $i \in [k+1]$. Let C be the connected component in $\widehat{\mathcal{C}}$ containing w. Notice that $c_1, c_2 \in M'$ as $(c_1, w), (c_2, w) \in E(G)$, $V(\mathbb{O}) \cap (V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A})) = \{w\}$, and $N_G(w) \subseteq M' \cup C$. Let us first argue that there exists an occurrence of $\mathsf{inc}(w)$ in $\mathbf{V}_{\{c_1,c_2\}} \setminus \widehat{\mathbf{V}}^{i-1}_{\{c_1,c_2\}}$. To this end, note that as w is the shallow terminal of $\mathbb{O}$, it is adjacent to c_1 and c_2, and therefore $w \in A_{\{c_1,c_2\}}$. Moreover, because $w \notin \widehat{A}$, there exists an occurrence of $\mathsf{inc}(w)$ that does not belong to $\widehat{\mathbf{V}}^{k+1}_{\{c_1,c_2\}}$, which implies that there exists an occurrence of $\mathsf{inc}(w)$ in $\mathbf{V}_{\{c_1,c_2\}} \setminus \widehat{\mathbf{V}}^{i-1}_{\{c_1,c_2\}}$.

As we have shown that $\mathsf{inc}(w)$ in $\mathbf{V}_{\{c_1,c_2\}} \setminus \widehat{\mathbf{V}}^{i-1}_{\{c_1,c_2\}}$, the fact that $\mathbf{B}^i_{\{c_1,c_2\}}$ is a basis for $\mathbf{V}_{\{c_1,c_2\}} \setminus \widehat{\mathbf{V}}^{i-1}_{\{c_1,c_2\}}$ implies that there exist vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t$ for some $t \in \mathbb{N}$ (in particular, $t \geq 1$) and nonzero coefficients $\lambda_1, \lambda_2, \ldots, \lambda_t$ such that $\lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \cdots + \lambda_t \mathbf{v}_t = \mathsf{inc}(w)$ over $\mathbb{F}_2^q$. Coefficient are from $\mathbb{F}_2$, so they are all necessarily 1. Thus, we have that

$$\mathbf{v}_1 + \mathbf{v}_2 + \cdots + \mathbf{v}_t = \mathsf{inc}(w) \text{ over } \mathbb{F}_2^q.$$

Denote $u_i = u_{\mathbf{v}_i}$ for all $i \in [t]$. Then, $\mathsf{inc}(u_1) + \mathsf{inc}(u_2) + \cdots + \mathsf{inc}(u_t) = \mathsf{inc}(w)$ over $\mathbb{F}_2^q$. In particular, $\mathsf{inc}^P(u_1) + \mathsf{inc}^P(u_2) + \cdots + \mathsf{inc}^P(u_t) = \mathsf{inc}^P(w)$ over $\mathbb{F}_2^q$, where P is the extended base of $\mathbb{O}$. This implies that $\sum_{i=1}^t \sum_{j=1}^q \mathsf{inc}^P(u_i)[j] = \sum_{j=1}^q \mathsf{inc}^P(w)[j] \mod 2$. (Note that since $V(\mathbb{O}) \cap (V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A})) = \{w\}$, the extended base is completely contained in $G[V(G) \setminus (V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A}))]$, and furthermore P is not covered by $\mathcal{W}$ by the premise of the lemma.) By Lemma 4.10 and since $N_G(w) \cap V(P) = \emptyset$ (because w is the shallow terminal of $\mathbb{O}$), we have that $\sum_{j=1}^q \mathsf{inc}^P(w)[j] = 1 \mod 2$. Thus, $\sum_{i=1}^t \sum_{j=1}^q \mathsf{inc}^P(u_i)[j] = 1 \mod 2$. This implies that there exists $i \in [t]$ such that $\sum_{j=1}^q \mathsf{inc}^P(u_i)[j] = 1 \mod 2$. However, by Lemma 4.10, this means that $N_G(u_i) \cap V(P) = \emptyset$. Moreover, we have that $u_i \in A_{\{c_1,c_2\}}$ because u_i is associated with the vector $\mathbf{v}_i$ which belongs to $\mathbf{B}^i_{\{c_1,c_2\}}$. Hence, $G[(V(\mathbb{O}) \setminus \{w\}) \cup \{u_i\}]$ is an AW. This completes the proof. ☐

Due to the definition of $\widehat{A}$, as a direct corollary to Lemma 4.11 we have the following result.

Corollary 4.1. *Let $w \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A})$, and $\mathbb{O}$ be an AW that is not covered by $\mathcal{W}$ such that $V(\mathbb{O}) \cap (V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A' \cup \widehat{A})) = \{w\}$ and w is the shallow terminal of $\mathbb{O}$. Then, there exists a set $\tilde{A} \subseteq \widehat{A}$ of size $k+1$ such that for each $u \in \tilde{A}$, $G[(V(\mathbb{O}) \setminus \{w\}) \cup \{u\}]$ is an obstruction.*

We are now ready to conclude the proof of Lemma 4.2 and thereby this section.

Proof of Lemma 4.2. Towards the proof, first note that if the condition of Reduction Rule 4.3 applies, then we are clearly done—indeed, in this case we output an instance (G', k) equivalent to (G, k) where G' is a strict subgraph of G. Thus, we next suppose that this rule has been applied exhaustively. Then, our output is the set $B = A^\star \cup A' \cup \widehat{A}$. By Observations 4.3 and 4.5, and by Lemma 4.9, we have that $|B| \leq |A^\star| + |A'| + |\widehat{A}| \leq (k+1)|M'|^2 + (k+1)|M'|^4 + 2(k+1)^2|M'|^6 \leq 4(k+1)^2|M'|^6$ as desired.

Let $S \subseteq V(G)$ be some arbitrary set of size at most k. We claim that the following property holds: If there exists an obstruction $\mathbb{O}$ for G that is not covered by $\mathcal{W}$ and such that $V(\mathbb{O}) \cap S = \emptyset$, then there exists an obstruction $\mathbb{O}'$ for G such that $V(\mathbb{O}') \cap S = \emptyset$ and $V(\mathbb{O}') \cap (V(\widehat{\mathcal{C}}) \setminus B) = \emptyset$. Clearly, if there does not exist any obstruction $\mathbb{O}$ for G that is not covered by $\mathcal{W}$ and such that $V(\mathbb{O}) \cap S = \emptyset$, then our proof is complete. Hence, we next suppose that such an obstruction exists, and we let $\mathbb{O}'$ be such a minimal obstruction that minimizes $|V(\mathbb{O}') \cap (V(\widehat{\mathcal{C}}) \setminus B)|$. We claim that for this obstruction $\mathbb{O}'$, it holds that $V(\mathbb{O}') \cap (V(\widehat{\mathcal{C}}) \setminus B) = \emptyset$, which would complete the proof. Suppose, by way of contradiction, that this claim is false. Then, as $V(\mathcal{C}^\star) \subseteq B$, there exists $C \in \widehat{\mathcal{C}} \setminus \mathcal{C}^\star$ and $v \in V(C)$ such that $v \in V(\mathbb{O}')$. By Lemma 4.7, $|V(\mathbb{O}) \cap V(C)| = 1$ and $\mathbb{O}'$ is an AW where v is a terminal.

Let us first suppose that v is not the shallow terminal of $\mathbb{O}'$. Then, by Lemma 4.8, there exist $(k+1)$ vertices $u \in A'$ such that $G[(V(\mathbb{O}') \setminus \{v\}) \cup \{u\}]$ is an obstruction. However, as $|S| \leq k$, this means that there exists $u \in A' \setminus S$ such that $G[(V(\mathbb{O}') \setminus \{v\}) \cup \{u\}]$ is an obstruction. As $A' \subseteq B$ and $G[(V(\mathbb{O}') \setminus \{v\}) \cup \{u\}]$ has fewer vertices from $V(\widehat{\mathcal{C}}) \setminus B$ than $\mathbb{O}'$, we have reached a contradiction to the choice of $\mathbb{O}$.

As the choice of v was arbitrary, we derive that $V(\mathbb{O}') \cap (V(\widehat{\mathcal{C}}) \setminus B)$ contains exactly one vertex, which we denote by w, that is the shallow terminal of $\mathbb{O}'$. In this case, by Corollary 4.1, there exist $(k+1)$ vertices $u \in \widehat{A}$ such that $G[(V(\mathbb{O}) \setminus \{w\}) \cup \{u\}]$ is an obstruction. However, as $|S| \leq k$, this means that there exists $u \in \widehat{A} \setminus S$ such that $G[(V(\mathbb{O}') \setminus \{w\}) \cup \{u\}]$ is an obstruction. As $\widehat{A} \subseteq B$ and $G[(V(\mathbb{O}') \setminus \{w\}) \cup \{u\}]$ has no vertices from $V(\widehat{\mathcal{C}}) \setminus B$, we have again reached a contradiction to the choice of $\mathbb{O}$. This completes the proof. ☐

4.1 Bounded Intersection Two Families Lemma

At the heart of our marking scheme to handle shallow terminals is in fact the special case of Lemma 1.1 where $c = 2$. Indeed, viewing this case in a more abstract manner, let us give a rough description of the relation between it and the statement of Lemma 1.1. For all $c_1, c_2 \in M'$, we have sets $A_1, A_2, \ldots, A_t$ and $B_1, B_2, \ldots, B_t$, that are defined as follows. First, the universe is the set of all vertices and pairs of vertices in N. Second, let W denote a set of vertices $w \in V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A')$ such that **(i)** w is adjacent to c_1 and c_2, and **(ii)** w has at least one induced path in $G[N]$, say P_w, which contains no vertex adjacent to w, and so that the two following properties hold:

- For all distinct $w, w' \in W$, w is adjacent to at least one vertex on $P_{w'}$.
- For every induced path P in $G[N]$ that has no vertex adjacent to some vertex in $V(\widehat{\mathcal{C}}) \setminus (A^\star \cup A')$, there also exists a vertex in W that is not adjacent to any vertex on P.

These properties mean, in a sense, that W is a minimal set that "covers" all induced paths in $G[N]$ that can potentially create AWs together with c_1 and c_2 as centers. Then, $t = |W|$, and denote $W = \{w_1, w_2 \ldots, w_t\}$. For every vertex $w_i \in W$, we create the new set A_i, which contains all the neighbors of w_i in N, and the new set B_i, which is equal to $V(P_{w_i})$. Clearly, for all $i \in [t]$, $A_i \cap B_i = \emptyset$, and due to Observation 4.6, for all distinct $i, j \in [t]$, $|A_i \cap B_i| \in \{1, 2\}$.

Let us now turn to the proof of Lemma 1.1. For convenience, let us restate it.

LEMMA 1.1. (Bounded Intersection Two Families) *Let $A_1, \ldots, A_m$ and $B_1, \ldots, B_m$ be families over a universe U such that (i) for every $i \leq m$, $A_i \cap B_i = \emptyset$, and (ii) for every $j \neq i$, $|A_i \cap B_j| \in \{1, \ldots, c\}$. Then $m \leq \sum_{t=0}^{c} \binom{|U|}{t}$.*

Proof. Let $|U| = n$ and let $d = \sum_{t=0}^{c} \binom{n}{t}$. Let D be the set of all subsets of U of size at most c (including the empty set). Note that we have $|D| = d$. Fix a bijection between D and $\{1, 2, \ldots, d\}$. We construct an incidence vector $\mathbf{v}_i$ for each set A_i, where $\mathbf{v}_i$ is indexed by the subsets of U of size up to c. More precisely, we have a vector $\mathbf{v}_i \in \{0,1\}^d$, where $\mathbf{v}_i[X] = 1$ if and only if $X \subseteq A_i$. Let us note that $\mathbf{v}_i[\emptyset] = 1$ for all $1 \leq i \leq m$. We consider these vectors as elements of the vector space $\mathbb{F}_2^d$. Similarly, we construct vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ for each set $B_1, B_2, \ldots, B_m$. We first claim that for every $i \in [m]$, we have $\mathbf{v}_i \cdot \mathbf{u}_i = 1$. This follows from the fact that $A_i \cap B_i = \emptyset$. We next claim that, for each $i, j \in [m]$, where $i \neq j$, we have $\mathbf{v}_i \cdot \mathbf{u}_j = 0$. This follows from the following observation. Let $C_{ij} = A_i \cap B_j$. Then, as $|C_{ij}| \in [c]$, we have that $2^{C_{ij}} \subseteq D$, where $2^{C_{ij}}$ denotes the collection of all subsets of C_{ij}. Now, observe that $\mathbf{v}_i[X]\mathbf{u}_j[X] = 1$ if and only if $X \subseteq C_{ij}$. As $|2^{C_{ij}}|$ is an even number (greater than or equal to 2), it follows that $\mathbf{v}_i \cdot \mathbf{u}_j = 0$ over the field $\mathbb{F}_2$.

Now suppose that $m > d$. Then the collection $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ is not linearly independent in $\mathbb{F}_2^d$. Hence, there is a vector, say $\mathbf{v}_m$, such that $\mathbf{v}_m = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_{m-1}\mathbf{v}_{m-1}$, where $\alpha_j \in \mathbb{F}_2$ for each $j \in [m-1]$. We claim that there is a vector $\mathbf{v}_i$ such that $\mathbf{v}_i \cdot \mathbf{u}_m = 1$ for some $i \in [m-1]$. This follows from the following equation.

$$\mathbf{v}_m \cdot \mathbf{u}_m = (\sum_{j=1}^{m-1} \alpha_j \mathbf{v}_j) \cdot \mathbf{u}_m$$
$$\implies 1 = \sum_{j=1}^{m-1} \alpha_j (\mathbf{v}_j \cdot \mathbf{u}_m)$$

However, this is a contradiction. Hence, $m \leq d$. This concludes the proof of this lemma. $\square$

5 Bounding the Maximum Size of a Clique of Non-module Components

Let $\eta = 2^{10} \cdot 4(k+5)\binom{|M|}{10}$. Recall that $\mathcal{C}$ is the set of connected components of $G - M$, $\mathcal{D}$ is the set of connected components in $\mathcal{C}$ that are modules, and $\overline{\mathcal{D}} = \mathcal{C} \setminus \mathcal{D}$. Let $(\mathbb{P}, \beta)$ be a clique path of $G[V(\overline{\mathcal{D}})]$, $V(\mathbb{P}) = \{x_1, x_2, \ldots, x_t\}$, and for each $i \in [t]$ we let $B_i = \beta(x_i)$. Furthermore, let $\beta(\mathbb{P}) = \cup_{i=1}^{t} \beta(x_i)$. Let B_i be a bag such that $|B_i| > \eta$. Towards bounding the size of B_i, we mark some of the vertices in B_i, and delete all the unmarked vertices in B_i from G. In fact, in a step we *only delete* one unmarked vertex, and then repeat the whole kernelization algorithm on the reduced instance. In the following, we describe the precise marking procedure.

Marking Scheme. To define our marking scheme, we first introduce some notations. We define two functions namely, $\mathsf{id}_\ell^i, \mathsf{id}_r^i : B_i \rightarrow [t]$. Intuitively, these functions denote how far or close a vertex appears in the bags that are to the left and right of B_i, respectively. For a vertex $v \in B_i$, $\mathsf{id}_\ell^i(v)$ is the smallest integer $x \in [t]$ such that $v \in B_x$, and $\mathsf{id}_r^i(v)$ is the largest integer $y \in [t]$ such that $v \in B_y$. Note that for each $v \in B_i$, we have $\mathsf{id}_\ell^i(v) \leq i \leq \mathsf{id}_r^i(v)$. A frame $\mathbb{F} = (X, Y)$ in G is a pair of vertex subsets, such that $X \subseteq M$ of size at most 10 and $Y \subseteq X$. A vertex $v \in V(G)$ is said to *fit* a frame $\mathbb{F} = (X, Y)$ if $N_G(v) \cap X = Y$. We now move to the construction of the set $H_i \subseteq B_i$, of marked vertices. For each frame $\mathbb{F}$ in G, we create four sets $L_{\mathsf{far}}^{\mathbb{F},i}, L_{\mathsf{cls}}^{\mathbb{F},i}, R_{\mathsf{far}}^{\mathbb{F},i}, R_{\mathsf{cls}}^{\mathbb{F},i} \subseteq B_i$ of marked vertices each of size as most $k+5$ (and add these vertices to H_i) as follows.

- We create the set $L_{\mathsf{far}}^{\mathbb{F},i}$ as follows. Let W be the set of unmarked vertices in B_i, that fit the frame $\mathbb{F}$. If $|W| \leq k+5$, then add all the vertices in W to $L^{\mathbb{F},i}_{\mathsf{far}}$. Else, let $W_{\mathsf{low}} \subseteq W$ be the set of $k+5$ vertices with lowest id^i_ℓ values among the vertices in W. Add W_{low} to $L^{\mathbb{F},i}_{\mathsf{far}}$.

- We create the set $L^{\mathbb{F},i}_{\mathsf{cls}}$ as follows. Let W be the set of unmarked vertices in B_i, that fit the frame $\mathbb{F}$. If $|W| \leq k+5$, then add all the vertices in W to $L^{\mathbb{F},i}_{\mathsf{cls}}$. Else, let $W_{\mathsf{high}} \subseteq W$ be the set of $k+5$ vertices with highest id^i_ℓ values among the vertices in W. Add W_{high} to $L^{\mathbb{F},i}_{\mathsf{cls}}$.
- We create the set $R^{\mathbb{F},i}_{\mathsf{far}}$ as follows. Let W be the set of unmarked vertices in B_i, that fit the frame $\mathbb{F}$. If $|W| \leq k+5$, then add all the vertices in W to $R^{\mathbb{F},i}_{\mathsf{far}}$. Else, let $W_{\mathsf{high}} \subseteq W$ be the set of $k+5$ vertices with highest id^i_r values among the vertices in W. Add W_{high} to $R^{\mathbb{F},i}_{\mathsf{far}}$.
- We create the set $R^{\mathbb{F},i}_{\mathsf{cls}}$ as follows. Let W be the set of unmarked vertices in B_i, that fit the frame $\mathbb{F}$. If $|W| \leq k+5$, then add all the vertices in W to $R^{\mathbb{F},i}_{\mathsf{cls}}$. Else, let $W_{\mathsf{low}} \subseteq W$ be the set of $k+5$ vertices with lowest id^i_r values among the vertices in W. Add W_{low} to $R^{\mathbb{F},i}_{\mathsf{cls}}$.

Notice that $|H_i| \leq 2^{10} \cdot 4(k+5)\binom{|M|}{10} = \eta$. Before proceeding further, we observe (Observation 5.1 and 5.2) certain useful properties regarding a frame $\mathbb{F}$ to which $v \in B_i \setminus H_i$ fits and the vertices in $L^{\mathbb{F},i}_{\mathsf{far}}, R^{\mathbb{F},i}_{\mathsf{far}}, L^{\mathbb{F},i}_{\mathsf{cls}}$, and $R^{\mathbb{F},i}_{\mathsf{cls}}$.

Observation 5.1. *For a frame $\mathbb{F} = (X, Y)$ to which v fits and a vertex $w \in N_G(v)$ the following holds.*

- *If $w \in Y$, then $L^{\mathbb{F},i}_{\mathsf{far}} \cup R^{\mathbb{F},i}_{\mathsf{far}} \subseteq N_G(w)$.*
- *If $w \in V(G) \setminus M$, then at least one of $L^{\mathbb{F},i}_{\mathsf{far}} \setminus \{w\} \subseteq N_G(w)$ or $R^{\mathbb{F},i}_{\mathsf{far}} \setminus \{w\} \subseteq N_G(w)$ holds.*

Proof. In the first case, it follows from the definition that $L^{\mathbb{F},i}_{\mathsf{far}} \cup R^{\mathbb{F},i}_{\mathsf{far}} \subseteq N_G(w)$. Now we prove the second part of the observation. First, consider the case when both v and w belong to B_i. In this case second claim holds, because B_i is a clique, $L^{\mathbb{F},i}_{\mathsf{far}} \subseteq B_i$ and $R^{\mathbb{F},i}_{\mathsf{far}} \subseteq B_i$. So let us assume that $w \notin B_i$. However, $w \in N_G(v)$ and hence both v and w lie in the same bag, say B_j, on the clique path $\mathbb{P}$. Since the bags in which w is present occur consecutively on $\mathbb{P}$, we have that all these bags either appear left of B_i or right of B_i. Let us consider the case when all the bags containing w appear left of B_i. The other case when all the bags containing w appear right of B_i is symmetric. We will show that $L^{\mathbb{F},i}_{\mathsf{far}} \setminus \{w\} \subseteq N_G(w)$. Towards this we will show that for every $x \in^{\mathbb{F},i}_{\mathsf{far}} \setminus \{w\}$, there exists a bag that contains both x and w. For a vertex z, let s_z denote the leftmost bag on $\mathbb{P}$ in which z appears and e_z denote the rightmost bag on $\mathbb{P}$ in which z appears. Recall that v is an unmarked vertex in B_i and thus, $s_x \leq s_v \leq i \leq e_x$. Furthermore, we know that $s_x \leq j < i$. This implies that x also belongs to B_j. Hence, we have shown that $L^{\mathbb{F},i}_{\mathsf{far}} \setminus \{w\} \subseteq N_G(w)$. This concludes the proof. □

Observation 5.2. *For a frame $\mathbb{F} = (X, Y)$ to which v fits and a vertex $w \notin N_G(v)$ the following holds.*

- *If $w \in X \setminus Y$, then $(L^{\mathbb{F},i}_{\mathsf{cls}} \cup R^{\mathbb{F},i}_{\mathsf{cls}}) \cap N_G(w) = \emptyset$.*
- *If $w \in V(G) \setminus M$, then at least one of $L^{\mathbb{F},i}_{\mathsf{cls}} \cap N_G(w) = \emptyset$ or $R^{\mathbb{F},i}_{\mathsf{cls}} \cap N_G(w) = \emptyset$ holds.*

Proof. In the first case, it follows from the definition that $(L^{\mathbb{F},i}_{\mathsf{cls}} \cup R^{\mathbb{F},i}_{\mathsf{cls}}) \cap N_G(w) = \emptyset$. In the second case, if $w \notin V(\mathcal{D})$ then the claim trivially holds. Otherwise, v and w lie in the clique path $\mathbb{P}$. Since $w \notin N_G(v)$, there is no bag which contains both v and w, and $v \in B_i$. Either w appears only in the bags (strictly) to the left of B_i, in which case v being an unmarked vertex implies that $L^{\mathbb{F},i}_{\mathsf{cls}} \cap N_G(w) = \emptyset$. On the other hand, if w appears only in the bags (strictly) to the right of B_i, we have $R^{\mathbb{F},i}_{\mathsf{cls}} \cap N_G(w) = \emptyset$. □

Next, we give a reduction rule that deletes unmarked vertices from B_i in G.

Reduction Rule 5.1. *Let v be a vertex in $B_i \setminus H_i$. Delete v from G i.e., the resulting instance is $(G - \{v\}, k)$.*

Lemma 5.1. *Reduction Rule 5.1 is safe.*

The proof of Lemma 5.1 is very long and requires a long detailed case analysis and will appear in the full version of the paper. We note that using Lemma 5.1, we immediately obtain the following lemma.

Lemma 5.2. *If Reduction Rule 5.1 is not applicable, then for each $j \in [t]$, we have $|B_j| \leq \eta$.*

Proof. Follows from the safeness of Reduction Rule 5.1 (Lemma 5.1) and the fact that $|H_j| \leq \eta$, for each $j \in [t]$. □

6 Bounding the Length of a Clique Path

Let us first recall the various sets we are dealing with. Let (G, k) be an instance of IVD.

- A $(k+2)$-necessary family $\mathcal{W} \subseteq 2^M$ along with a solution M that is 9-redundant with respect to $\mathcal{W}$. In fact, $\mathcal{W} \subseteq 2^M$.
- Every set in $\mathcal{W}$ has size at least 2.

- $\mathcal{C}$ is the set of connected components of $G - M$, $\mathcal{D}$ is the set of connected components in $\mathcal{C}$ that are modules, and $\overline{\mathcal{D}} = \mathcal{C} \setminus \mathcal{D}$. We know that $|V(\mathcal{D})| \leq k^{\mathcal{O}(1)}$ and $|\overline{\mathcal{D}}| \leq k^{\mathcal{O}(1)}$.
- Every maximal clique (and hence every clique) in $G - M$ has size bounded by η.

Let us now turn to the problem of bounding the sizes of non-module components. Observe that to bound this it is sufficient to "bound the length of the clique path'' of a non-module component. This together with the fact that each maximal clique is bounded will lead to the desired result. Our approach mirrors that of [1, 34], but requires additional structural observations corresponding to interval graphs and its obstructions [27, 7]. Each non-module component is a clique path in $G - M$, where M is a 9-redundant modulator.

Let $\mathbb{K} = (K, \beta)$ be a clique path of a non-module component C, where $V(K) = \{x_1, x_2, \ldots, x_t\}$, and for each $i \in [t]$ we let $B_i = \beta(x_i)$. We will refer to the sets B_i, $1 \leq i \leq t$, as the *bags* in $\mathbb{K}$. Any bag B_i in the clique path $\mathbb{K}$ has at most $\eta = 2^{10} \cdot 4(k+5)\binom{|M|}{10}$ vertices (because every maximal clique in $G-M$ has size bounded by η). We let $\beta(\mathbb{K}) = \cup_{i=1}^{t} \beta(x_i)$. Furthermore, for a subpath K' of K, by $\mathbb{K}' = (K', \beta')$ we denote the sub-clique path induced by K'. That is, for $x \in V(K')$, $\beta'(x) = \beta(x)$. Moreover, by $\beta(\mathbb{K}')$ we denote the set $\cup_{x \in V(K')} \beta(x)$. Note that there is a vertex in M that has a neighbor as well as a non-neighbor in C.

In this section, we consider the problem of reducing the number of bags in $\mathbb{K}$. Towards our goal, we will devise a collection of "marking schemes" that mark some polynomially (in k) many bags in $\mathbb{K}$, such that the obstructions are "well behaved" in the region between any two consecutive marked bags. In particular, our marking schemes ensure that if any obstruction intersects an unmarked region of the clique path, then the intersection is an *induced path*. Then, we design reduction rules that "preserve" a minimum separator of the unmarked region. More precisely, we identify an *irrelevant vertex* or an *irrelevant edge*, and then delete it or contract it in the graph. The correctness of these reduction rules follows from the structural properties ensured by the marking schemes.

Let us now define few notations that will be required in this section. Note that these notations apply to $\mathbb{K} = (K, \beta)$ as well as any sub-clique path of it. We fix an ordering (from left to right) of the bags of $\mathbb{K}$, which is given by the path K of the clique path $\mathbb{K}$. We will maintain a set of bags $\mathcal{B}$ in $\mathbb{K}$, which we will call *marked bags*. Initially, $\mathcal{B} = \emptyset$, and we will add some carefully chosen bags in $\mathbb{K}$ to it, as we proceed.

1. For two bags B_ℓ and B_r in $\mathbb{K}$, by $\mathbb{K}[B_\ell, B_r] = (K', \beta')$ we denote the sub-clique path of $\mathbb{K}$ between B_ℓ and B_r (including B_ℓ and B_r).
2. We say that a vertex $v \in \beta(\mathbb{K})$ is a *marked vertex* if there is a marked bag that contains it, otherwise it is an *unmarked vertex*.
3. We say that two marked bags B, B' are *consecutive* if $\mathbb{K}[B, B']$ contains no marked bags other than B and B'.
4. We say that two (distinct) bags B, B in $\mathbb{K}$ are *adjacent* if there is no other bag that lies between them, i.e. $\mathbb{K}[B, B']$ has only two bags, namely, B and B'.
5. For a bag B in $\mathbb{K}$, B^{-1} and B^{+1} denote the bags adjacent to B on its left and right, respectively.

6.1 Partition into Manageable Clique Paths In this section, we partition the clique path $\mathbb{K}$ into a collection of so called "manageable clique paths", which are well structured with respect to the set M. We will construct a set of marked bags, denoted by $\mathbb{K}(M)$, based on the edges between the vertices in $\beta(\mathbb{K})$ and M. Let us initialize $\mathbb{K}(M)$ as the set containing the first and last bags of $\mathbb{K}$. We begin by stating a property of interval graphs, which will be useful later.

Observation 6.1. *Let H be an interval graph and let H' be the graph obtained by one of the following operations.*

(a) For $v \in V(H)$, $H' = H - \{v\}$.
(b) For $(u, v) \in E(H)$, $H' = H/(u, v)$.

Then H' is an interval graph. Furthermore, the size of any clique in H' is upper-bounded by the size of a maximum clique in H.

The above observation follows from the definition of interval graphs and their interval representation [27]. In particular, statement (b) follows from the observation that an interval representation of $H/(u, v)$ can be obtained by taking an interval representation of H and "merging" the intervals of u and v.

In the following, we will define (auxiliary) graphs that will be helpful in obtaining some useful bags in $\mathbb{K}$. To this end, consider a vertex $m \in M$. Let H_m be the bipartite graph with vertex bipartition $N_G(m) \cap \beta(\mathbb{K})$ and $\beta(\mathbb{K}) \setminus N_G(m)$, where $u \in N_G(m) \cap \beta(\mathbb{K})$ and $v \in \beta(\mathbb{K}) \setminus N_G(m)$ are adjacent in H_m if and only if $(u, v) \in E(G)$. Next, we prove the following lemma about the graph H_m. (Recall that η is an upper bound on the size of any clique in $G - M$.)

Lemma 6.1. *For $m \in M$, let Y_m be a maximum matching in H_m. Then $|Y_m| \leq 2\eta$.*

Proof. Suppose, towards a contradiction, that $|Y_m| > 2\eta$. Let T be the graph obtained from $G[\beta(\mathbb{K})]$ by contracting all the edges in Y_m. Additionally, for each edge (u, v) in Y_m, let w_{uv} be the vertex resulting from its contraction. Recall that $G - M$ is an interval graph of maximum clique size at most η, which together with Observation 6.1 implies that both $G[\beta(\mathbb{K})]$ and T are also interval graphs, and that the maximum size of a clique in these graphs is upper bounded by η. Next, let $\tilde{T}$ be the graph $T[\{w_{uv} \mid (u, v) \in Y_m\}]$. We note that the definition of $\tilde{T}$ relies on the fact that Y_m is a matching in H_m, and thus it has $|Y_m| > 2\eta$ many vertices. From the construction of $\tilde{T}$ and Observation 6.1, it follows that $\tilde{T}$ is also an interval graph and that the size of any clique in $\tilde{T}$ is bounded by η. Interval graphs are perfect graphs, and on a perfect graph $\mathcal{G}$ we know that $\omega(\mathcal{G})\alpha(\mathcal{G}) \geq |V(\mathcal{G})|$, where $\omega(\mathcal{G})$ and $\alpha(\mathcal{G})$ denote the size of a maximum clique and a maximum independent set in $\mathcal{G}$, respectively [45] (or Theorem 3.3 [27]). This implies that there is an independent set in $\tilde{T}$ of size at least $|Y_m|/\eta > 2$. Consider an independent set of size 3 in $\tilde{T}$, and the corresponding edges of the matching Y_m. It follows that these three edges and the vertex m form a long claw, $\mathbb{O}$ in G, which is an obstruction of size 7. Since Reduction Rule 3.1 is not applicable, each set in $\mathcal{W}$ is of size at least 2. Moreover, $|V(\mathbb{O}) \cap M| = 1$. Therefore, $\mathbb{O}$ is not covered by $\mathcal{W}$. But then, since M is a 9-redundant solution each obstruction in G which is not covered by $\mathcal{W}$ must contain at least 10 vertices from M. Thus, we deduce that $|Y_m| > 2\eta$ cannot hold. □

For each $m \in M$, we compute a maximum matching Y_m in the graph H_m. Then for each edge in Y_m we pick a bag in $\mathbb{K}$ that contains this edge and add it to $\mathbb{K}(M)$. Let us observe that we have added at most $2\eta|M|$ bags to $\mathbb{K}(M)$. Before proceeding further, we add some more bags to $\mathbb{K}(M)$ that give us some additional structural properties. Next, we state the following observation, which will be useful in designing one of our marking schemes for bags in $\mathbb{K}$.

Observation 6.2. *Let $m_1, m_2 \in M$ be (distinct) vertices such that $\{m_1, m_2\} \notin \mathcal{W}$ and $(m_1, m_2) \notin E(G)$. Then, $(N_G(m_1) \cap N_G(m_2)) \setminus M$ induces a clique in G.*

Proof. This observation is the special case of Lemma 4.4 with $M' = M, u = m_1, v = m_2$ and $u, v \in M$. □

Next, consider (distinct) $m_1, m_2 \in M$, such that $\{m_1, m_2\} \notin \mathcal{W}$ and $(m_1, m_2) \notin E(G)$. Let $B(m_1, m_2)$ be a bag in $\mathbb{K}$, such that $(N_G(m_1) \cap N_G(m_2)) \cap \beta(\mathbb{K}) \subseteq B(m_1, m_2)$. We note that the existence of $B(m_1, m_2)$ is guaranteed from Observation 6.2. We add $B(m_1, m_2)$ to the set $\mathbb{K}(M)$. We are now ready to state our first bag-marking scheme.

Marking Scheme I. Add all the bags in $\mathbb{K}(M)$ to $\mathcal{B}$.

Note that $|\mathbb{K}(M)|$ is at most $2\eta|M| + |M|^2 + 2$. This bound is obtained because *(i)* $\mathbb{K}(M)$ contains the first and last bag of $\mathbb{K}$, *(ii)* at most 2η bags in $\mathbb{K}$ were added corresponding to the matching Y_m for each $m \in M$ (and H_m), and *(iii)* for (distinct) $m_1, m_2 \in M$, such that $\{m_1, m_2\} \notin \mathcal{W}$ and $(m_1, m_2) \notin E(G)$, we added a bag to $\mathbb{K}(M)$. Thus, using Marking Scheme I, we have marked at most $\boxed{2\eta|M| + |M|^2 + 2 < 4\eta|M|}$ bags in $\mathbb{K}$. Here, we used the fact that $\eta \geq |M|$.

We apply *six marking schemes*, similar in spirit to the above, and we mark $k^{\mathcal{O}(1)}$ vertices in total. If there is an unmarked vertex then we can either find an "irrelevant vertex" to delete or an "irrelevant edge" to contract. In either case, we reduce the number of vertices in the graph. The above requires carefully exploiting the structure of the input instance and doing exhaustive case distinctions. Finally, when none of the devised reduction rules are applicable, we obtain a kernel whose size is bounded by $k^{\mathcal{O}(1)}$. We leave the full details for the longer version of the paper.

7 Conclusion

In this paper, we proved that the IVD problem admits a polynomial kernel. We remark that the degree in the polynomial that bounds the kernel size can be improved to be about a 100 at the cost of significantly more involved arguments. In particular, this can be done by considering a solution M of lower redundancy and far more involved case analysis for bounding the clique size and clique paths of $G - M$ in Sections 5 and 6. However, obtaining a kernel of size around $O(k^{10})$ will require new ideas. We leave this as an interesting open problem. We also believe that our techniques and methods, especially the Two Families Lemma (Lemma 1.1), will be useful in other algorithmic applications.

References

[1] A. Agrawal, D. Lokshtanov, P. Misra, S. Saurabh, and M. Zehavi, *Feedback vertex set inspired kernel for chordal vertex deletion*, in Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA), 2017, pp. 1383–1398. 3, 4, 17

[2] ———, *Feedback vertex set inspired kernel for chordal vertex deletion*, CoRR, abs/1707.04917 (2017). 4

[3] A. Agrawal, D. Lokshtanov, A. E. Mouawad, and S. Saurabh, *Simultaneous Feedback Vertex Set:*

A Parameterized Perspective, in STACS 2016, vol. 47 of LIPIcs, 2016, pp. 7:1–7:15. 4

[4] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, *A unified approach to approximating resource allocation and scheduling*, J. ACM, 48 (2001), pp. 1069–1090. 1

[5] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin, *On problems without polynomial kernels*, Journal of Computer and System Sciences, 75 (2009), pp. 423–434. 1

[6] B. Bollobás, *On generalized graphs*, Acta Mathematica Hungarica, 16 (1965), pp. 447–452. 3

[7] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph classes: a survey*, SIAM, 1999. 1, 17

[8] Y. Cao, *Unit interval editing is fixed-parameter tractable*, in ICALP, vol. 9134 of Lecture Notes in Computer Science, Springer, 2015, pp. 306–317. 4

[9] ———, *Linear recognition of almost interval graphs*, in SODA, 2016, 2016, pp. 1096–1115. 1, 2, 4, 6

[10] Y. Cao and D. Marx, *Interval deletion is fixed-parameter tractable*, ACM Trans. Algorithms, 11 (2015), pp. 21:1–21:35. 1, 4, 10

[11] ———, *Chordal editing is fixed-parameter tractable*, Algorithmica, 75 (2016), pp. 118–137. 4

[12] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015. 1, 5

[13] M. Cygan, Ł. Kowalik, and M. Pilipczuk, *Open problems from workshop on kernels*, Available at `http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf`, (2013). 2

[14] H. Dell and D. Marx, *Kernelization of packing problems*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2012, pp. 68–81. 1

[15] H. Dell and D. van Melkebeek, *Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses*, Journal of the ACM, 61 (2014), pp. 23:1–23:27. 1

[16] R. Diestel, *Graph Theory*, Springer-Verlag, Heidelberg, 4th ed., 2010. 4

[17] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013. 1, 5

[18] A. Drucker, *New limits to classical and quantum instance compression*, SIAM Journal on Computing, 44 (2015), pp. 1443–1479. 1

[19] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006. 1, 5

[20] F. V. Fomin, B. M. P. Jansen, and M. Pilipczuk, *Preprocessing subgraph and minor problems: When does a small vertex cover help?*, J. Comput. Syst. Sci., 80 (2014), pp. 468–495. 3

[21] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh, *Planar F-deletion: Approximation, kernelization and optimal FPT algorithms*, in FOCS, 2012. 1, 4

[22] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh, *Efficient computation of representative families with applications in parameterized and exact algorithms*, Journal of the ACM, 63 (2016), pp. 29:1–29:60. 3

[23] F. V. Fomin and S. Saurabh, *Kernelization methods for fixed-parameter tractability*, in Tractability, Cambridge Univ. Press, Cambridge, 2014, pp. 260–282. 1

[24] L. Fortnow and R. Santhanam, *Infeasibility of instance compression and succinct PCPs for NP*, Journal of Computer and System Sciences, 77 (2011), pp. 91–106. 1

[25] T. Fujito, *A unified approximation algorithm for node-deletion problems*, Discrete Appl. Math., 86 (1998), pp. 213–231. 1

[26] D. Fulkerson and O. Gross, *Incidence matrices and interval graphs*, Pacific journal of mathematics, 15 (1965), pp. 835–855. 5

[27] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980. 1, 5, 17, 18

[28] J. Guo and R. Niedermeier, *Invitation to data reduction and problem kernelization*, SIGACT News, 38 (2007), pp. 31–45. 1

[29] D. Hermelin, S. Kratsch, K. Soltys, M. Wahlström, and X. Wu, *A completeness theory for polynomial (turing) kernelization*, Algorithmica, 71 (2015), pp. 702–730. 1

[30] D. Hermelin and X. Wu, *Weak compositions and their applications to polynomial lower bounds for kernelization*, in Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2012, pp. 104–113. 1

[31] B. M. Jansen, *The power of data reduction: Kernels for Fundamental Graph Problems*, PhD thesis, Utrecht University, 2013. 2

[32] B. M. P. Jansen, D. Lokshtanov, and S. Saurabh, *A near-optimal planarization algorithm*, in SODA, 2014, pp. 1802–1811. 4

[33] B. M. P. Jansen and M. Pilipczuk, *Approximation and kernelization for chordal vertex deletion*, CoRR, abs/1605.03001 (2016). 2

[34] ———, *Approximation and kernelization for chordal vertex deletion*, in Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19, 2017, pp. 1399–1418. 3, 4, 17

[35] Y. Ke, Y. Cao, X. Ouyang, W. Li, and J. Wang, *Unit interval vertex deletion: Fewer vertices are relevant*, J. Comput. Syst. Sci., 95 (2018), pp. 109–121. 4

[36] D. Kendall, *Incidence matrices, interval graphs and seriation in archeology*, Pacific Journal of mathematics, 28 (1969), pp. 565–570. 1

[37] E. J. Kim and O. Kwon, *A polynomial kernel for block graph deletion*, in IPEC, vol. 43 of LIPIcs, 2015, pp. 270–281. 4

[38] E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar, *Linear kernels and single-exponential algorithms via protrusion decompositions*, ACM Trans. Algorithms, 12 (2016), pp. 21:1–21:41. 4

[39] S. Kratsch, *Recent developments in kernelization: A survey*, Bulletin of the EATCS, 113 (2014). 1, 5

[40] S. Kratsch and M. Wahlstrom, *Representative sets and irrelevant vertices: New tools for kernelization*, 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, 0 (2012), pp. 450–459. 3

[41] D. C. Lay, *Linear algebra and its applications*, Linear Algebra and Its Applications, Pearson/Addison-Wesley, 2006. 6

[42] C. Lekkeikerker and J. Boland, *Representation of a finite graph by a set of intervals on the real line*, Fundamenta Mathematicae, 51 (1962), pp. 45–64. 2

[43] J. M. Lewis and M. Yannakakis, *The node-deletion problem for hereditary properties is NP-complete*, J. Comput. Syst. Sci., 20 (1980), pp. 219–230. 1

[44] D. Lokshtanov, N. Misra, and S. Saurabh, *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, 2012, pp. 129–161. 1, 5

[45] L. Lovász, *A characterization of perfect graphs*, Journal of Combinatorial Theory, Series B, 13 (1972), pp. 95–98. 18

[46] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, J. ACM, 41 (1994), pp. 960–981. 1

[47] D. Marx, *A parameterized view on matroid optimization problems*, Theor. Comput. Sci., 410 (2009), pp. 4471–4479. 3

[48] D. Marx, *Chordal deletion is fixed-parameter tractable*, Algorithmica, 57 (2010), pp. 747–768. 3

[49] R. Niedermeier, *Invitation to fixed-parameter algorithms*, vol. 31 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2006. 1, 5

[50] M. Yannakakis, *Node-and edge-deletion NP-complete problems*, in STOC, New York, NY, USA, 1978, ACM, pp. 253–264. 1