A Polynomial Kernel for Deletion to Ptolemaic Graphs

Akanksha Agrawal
Indian Institute of Technology Madras, Chennai, India

Aditya Anand
Indian Institute of Technology Kharagpur, India

Saket Saurabh
Institute of Mathematical Sciences, Chennai, India
University of Bergen, Norway

Abstract

For a family of graphs $\mathcal{F}$, given a graph G and an integer k, the $\mathcal{F}$-Deletion problem asks whether we can delete at most k vertices from G to obtain a graph in the family $\mathcal{F}$. The $\mathcal{F}$-Deletion problems for all non-trivial families $\mathcal{F}$ that satisfy the hereditary property on induced subgraphs are known to be NP-hard by a result of Yannakakis (STOC'78). Ptolemaic graphs are the graphs that satisfy the Ptolemy inequality, and they are the intersection of chordal graphs and distance-hereditary graphs. Equivalently, they form the set of graphs that do not contain any chordless cycles or a gem as an induced subgraph. (A gem is the graph on 5 vertices, where four vertices form an induced path, and the fifth vertex is adjacent to all the vertices of this induced path.) The Ptolemaic Deletion problem is the $\mathcal{F}$-Deletion problem, where $\mathcal{F}$ is the family of Ptolemaic graphs. In this paper we study Ptolemaic Deletion from the viewpoint of Kernelization Complexity, and obtain a kernel with $\mathcal{O}(k^6)$ vertices for the problem.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Ptolemaic Deletion, Kernelization, Parameterized Complexity, Gem-free chordal graphs

Digital Object Identifier 10.4230/LIPIcs.IPEC.2021.1

Related Version *Full Version*: https://akanksha-agrawal.weebly.com/uploads/1/2/2/2/122276497/ptolemaic-deletion.pdf

Funding *Akanksha Agrawal*: Supported by New Faculty Initiation Grant (IIT Madras).
Saket Saurabh: Supported by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416) and Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.

1 Introduction

Graph modification problems are one of the central problems in Graph Theory, and vertex deletion is one of the most natural graph modification operations. For a family of graphs $\mathcal{F}$, $\mathcal{F}$-Deletion takes as input an n-vertex graph G and an integer k, and the objective is to determine whether we can remove a set of at most k vertices from G to obtain a graph in $\mathcal{F}$. Some of the classical examples of $\mathcal{F}$-Deletion are the NP-hard problems like Vertex Cover, Feedback Vertex Set, and Odd Cycle Transversal, corresponding to $\mathcal{F}$ being the family of edgeless graphs, forests, and bipartite graphs, respectively. Unfortunately, most of these $\mathcal{F}$-Deletion problems are NP-hard by a result of [19, 22]. Thus, they have received substantial attention in the algorithmic paradigms for coping with NP-hardness, including Approximation Algorithms and Parameterized Complexity.

In Parameterized Complexity each problem instance is accompanied by a parameter k. One of the central notions in parameterized complexity is that of *fixed parameter tractability* (FPT). A parameterized problem Π is FPT if given an instance (I, k) of Π, we can determine whether (I, k) is a YES-instance of Π in time bounded by $\mathcal{O}(f(k)|I|^{\mathcal{O}(1)})$, where f is some computable function of k and $|I|$ is the encoding length of I. One way to obtain an FPT algorithm for a (decidable) parameterized problem algorithm is to exhibit a *kernelization algorithm*, or *kernel*. A kernel for a problem Π is an algorithm that given an instance (I, k) of Π, runs in polynomial time and outputs an equivalent instance (I', k') of Π such that $|I'|, k'$ are both upper bounded by $g(k)$ for some computable function g. The function g is the *size* of the kernel, and if g is a polynomial function, then we say that the kernel is a *polynomial kernel*. A kernel for a decidable problem implies that it admits an FPT algorithm, but kernels are also very interesting in their own right, as they mathematically capture the efficiency of polynomial time pre-processing routines.

A graph is a *chordal* graph if it does not contain any induced cycle on at least four vertices. A graph G is *distance hereditary* if the distances between vertices in every connected induced subgraph of G are the same as in the graph G. *Ptolemaic graphs* are the graphs that are both chordal and distance hereditary. In this paper we study the $\mathcal{F}$-DELETION problem corresponding to the family of Ptolemaic graphs. Formally, we study the following problem.

PTOLEMAIC DELETION	**Parameter:** k
Input: A graph G with n vertices and a non-negative integer k.	
Question: Is there $X \subseteq V(G)$ such that $|X| \leqslant k$ and $G \setminus X$ is a Ptolemaic graph?	

We study the parameterized complexity of the PTOLEMAIC DELETION problem. Recently, Ahn et al. [7] obtained a constant-factor approximation algorithm for the weighted version of the problem. In their algorithm, for a given instance of PTOLEMAIC DELETION, they create an equivalent (and approximation preserving) instance of a special FEEDBACK VERTEX SET problem, which they call FEEDBACK VERTEX SET WITH PRECEDENCE CONSTRAINTS. With a minor modification to the well-known iterative-compression based FPT algorithm for FEEDBACK VERTEX SET (see, for example [9]), we can obtain a $c^k n^{\mathcal{O}(1)}$-time algorithm for FEEDBACK VERTEX SET WITH PRECEDENCE CONSTRAINTS. The above together with the reduction from PTOLEMAIC DELETION to the problem FEEDBACK VERTEX SET WITH PRECEDENCE CONSTRAINTS presented in [7] implies that PTOLEMAIC DELETION admits a simple $c^k n^{\mathcal{O}(1)}$-time FPT algorithm. We remark that both CHORDAL VERTEX DELETION [20] and DISTANCE HEREDITARY DELETION [13] are FPT, but their algorithms are more involved, compared to the elegant FPT algorithm that PTOLEMAIC DELETION admits.

Due to the existence of a simple single-exponential FPT algorithm for PTOLEMAIC DELETION, in this paper we focus on obtaining a polynomial kernel for the problem. In particular, we obtain the following result.

▶ **Theorem 1.** PTOLEMAIC DELETION *admits a kernel with* $\mathcal{O}(k^6)$ *vertices.*

We note that the kernelization complexity of CHORDAL VERTEX DELETION was a well-known open problem in the field of Parameterized Complexity. Jansen and Pilipczuk [16] designed the first polynomial kernel for the above problem, and shortly afterwards, Agrawal et al. [3] gave an improved kernel with $\mathcal{O}(k^{13})$ vertices. Kim and Kwon [17] showed that DISTANCE HEREDITARY DELETION admits a kernel of size $\mathcal{O}(k^{30} \log^5 k)$. We remark that the kernel we obtain has only $\mathcal{O}(k^6)$ vertices, which is much smaller than the kernels for both these problems. We also believe that our techniques can prove to be useful in obtaining/improving kernels for related classes of graphs.

Our Methods. We now describe the techniques we use in obtaining our kernel, for the given instance (G, k) of Ptolemaic Deletion. First, we compute an approximate solution S, using the result of Ahn et al. [7]. We further construct a strengthened version of the approximate solution called a redundant solution, D, introduced in [4]. Roughly speaking, redundant solutions allow us to, in a sense, forget about small obstructions. This simplifies many technical difficulties, while only maintaining a small set of vertices. We analyse the set of maximal cliques and bound the size of any maximal clique in the Ptolemaic graph $G \setminus (S \cup D)$ by $\mathcal{O}(k^3)$ by exploiting structural properties of Ptolemaic graphs and the property of "redundancy" ensured by the set D. To this end, we also use matchings in auxiliary graphs, to determine which vertices are "safe" to delete.

After bounding the sizes of maximal cliques in $G \setminus (S \cup D)$, we make use of a characterization of Ptolemaic graphs based on their inter-clique digraph [21]. In particular, Uehara and Uno [21] showed that a graph is Ptolemaic if and only if the underlying undirected graph of the inter-clique digraph of the given graph is a forest. We now use the concept of independence degree introduced in [3], and bound the independence degree of certain vertices, introducing an annotated version of the problem, similar to [3]. We then design a procedure which exploits the bounded independence degree and obtain a set $R \subseteq D$, so that the number of leaves in the undirected inter-clique digraph of the Ptolemaic graph $G \setminus (R \cup S)$ can be bounded by $\mathcal{O}(k^3)$. Every vertex of this forest, which we call bags, corresponds to a set of vertices which form a clique in $G \setminus (R \cup S)$. This, together with the fact that the size of a maximal clique is bounded in $G \setminus (R \cup S)$, gives us a bound on the number of vertices contained in degree-1 and degree $\geqslant 3$ bags in the (undirected) forest. Finally, we bound the number of vertices in degree-2 bags. In order to do this, we examine the structure of degree-2 paths (paths containing only degree 2 bags of the forest) in the inter-clique digraph. We first give several structural reduction rules and then show that we can safely replace "large" portions of the graph with smaller-sized graphs, if we maintain the size of a minimum separator in an augmented graph. Combining everything together, we obtain our kernel for Ptolemaic Deletion with $\mathcal{O}(k^6)$ vertices.

Related Works. As Ptolemaic graphs are hereditary on induced subgraphs, by the seminal result of Lewis and Yannakakis [22, 19] it follows that Ptolemaic Deletion is NP-complete. As mentioned earlier, very recently Ahn et al. [7] obtained a constant factor approximation algorithm for Ptolemaic Deletion. There have been several studies of the parameterized complexity for deletion to subclasses of chordal graphs and distance hereditary graphs. As mentioned previously, both Chordal Vertex Deletion [3, 20, 16] and Distance Hereditary Deletion [13, 17] admit FPT algorithms and polynomial kernels. Interval graphs are an important subclass of chordal graphs. Cao and Marx [8] showed that Interval Vertex Deletion admits an FPT algorithm. Recently, Agrawal et al. [4] obtained the first polynomial kernel for Interval Vertex Deletion with $\mathcal{O}(k^{1741})$ vertices. Block graphs are a subclass of Ptolemaic graphs, and Block Vertex Deletion is known to admit an FPT algorithm running in time $4^k n^{\mathcal{O}(1)}$ and a kernel with $\mathcal{O}(k^4)$ vertices [2]. Split graphs form a well-known subclass of chordal graphs, and Split Vertex Deletion is known to admit a $\mathcal{O}(1.2738^k k^{\mathcal{O}(\log k)} + n^{\mathcal{O}(1)})$-time FPT algorithm [10] and a kernel with $\mathcal{O}(k^2)$ vertices [1]. Another class of graphs which form a subclass of Ptolemaic graphs are 3-leaf power graphs. The corresponding deletion problem, 3-Leaf Power Deletion was shown to be FPT by [11, 5]. Recently, Ahn et al. [6] designed a polynomial kernel for 3-Leaf Power Deletion.

2 Preliminaries

Sets and Undirected Graphs. For $k \in \mathbb{N}$, we use $[k]$ as a shorthand for $\{1, 2, \dots, k\}$. Given an undirected graph G, we let $V(G)$ and $E(G)$ denote its vertex-set and edge-set, respectively. We let n denote the number of vertices in a graph G, whenever the context is clear. The *open neighborhood*, or simply the *neighborhood*, of a vertex $v \in V(G)$ is defined as $N_G(v) = \{w \mid \{v, w\} \in E(G)\}$. We extend the definition of neighborhood of a vertex to a set of vertices as follows. Given a subset $U \subseteq V(G)$, $N_G(U) = \bigcup_{u \in U} N_G(u) \setminus U$. We omit subscripts when the graph G is clear from the context. The *induced subgraph* $G[U]$ is the graph with vertex-set U and edge set $\{\{u, u'\} \mid u, u' \in U$, and $\{u, u'\} \in E(G)\}$. Moreover, we define $G \setminus U$ as the induced subgraph $G[V(G) \setminus U]$.

We define the distance $d(u, v)$ between two vertices $u, v \in V(G)$ as the length of the shortest path between them.

For a graph G and vertices $s, t \in V(G)$, where $s \neq t$, an *s-t separator* is a subset $U \subseteq V(G)$ such that $G \setminus U$ has no s-t path. It is well known that a minimum sized s-t separator (also called a *minimum s-t separator*) can be found in polynomial time using, for example, an algorithm for maximum flow, see for instance Chapter 7 of [18].

A *gem* is a graph on five vertices, with one vertex adjacent to each of the remaining four vertices which form an induced path.

A graph G is *chordal* if it does not contain a chordless cycle as an induced subgraph. G is *distance hereditary* if for every connected induced subgraph H of G and two distinct vertices $u, v \in V(H)$, the length of the shortest path between u and v in H is equal to the length of the shortest path between u and v in G.

Ptolemaic Graphs. A graph G is a *Ptolemaic graph* if for every four vertices u, v, w, x in the same connected component, $d(u, v)d(w, x) + d(u, x)d(v, w) \geqslant d(u, w)d(v, x)$.

▶ **Proposition 2** (Theorems 2.1,3.2 [15])**.** *Given a graph G, the following statements are equivalent: i) G is Ptolemaic, ii) G is both chordal and distance hereditary, and iii) G does not contain a gem or a chordless cycle as an induced subgraph.*

We call gems and chordless cycles *obstructions* and say that G contains an obstruction, if it has an obstruction as an induced subgraph.

Inter-Clique Digraphs. For a graph G, let $\mathcal{C}(G)$ be the set that contains i) all maximal cliques in G, and ii) non-empty intersections of (any number of) distinct maximal cliques in G. Consider the directed graph D_G which has a vertex v_C for each $C \in \mathcal{C}(G)$. For every $X, Y \in \mathcal{C}(G)$, where i) $X \subset Y$, and ii) there is no $W \in \mathcal{C}(G)$ such that $X \subset W$ and $W \subset Y$; we add an arc from v_Y directed towards v_X. For a directed graph T, $Und(T)$ denotes the underlying undirected graph (obtained by removing the directions associated with arcs in T). We make use of a characterization of Ptolemaic Graphs based on Inter-Clique Digraphs by Uehara and Uno[21].

▶ **Proposition 3** (Theorem 5,8 [21])**.** *A graph G is Ptolemaic if and only if $Und(D_G)$ is a forest. Moreover, the inter-clique digraph of a Ptolemaic graph can be computed in linear time.*

For a Ptolemaic graph G, let T_G denote its inter-clique digraph. We refer to the vertices in T_G (and $Und(T_G)$) as *bags*, to avoid confusions. A *leaf* of T_G is a leaf in $Und(T_G)$. For a bag $B \in T_G$, denote the associated set of vertices in G in the bag B by $V(B)$.

In Weighted Ptolemaic Deletion, we are given a graph G and a weight function $w : V \to \mathbb{R}^+ \cup \{0\}$, and the objective is to compute a minimum weight subset $X \subseteq V(G)$ such that $G \setminus X$ is a Ptolemaic graph.

For a graph G, we say that a set $X \subseteq V(G)$ is a *solution* for G if $G \setminus X$ is a Ptolemaic graph. By $\mathsf{opt}(G)$, we denote the size of a minimum sized solution for G. Moreover, if we are given a weight function $w : V(G) \to \mathbb{R}^+ \cup \{0\}$, then $\mathsf{opt}(G)$ denotes the weight of a minimum weight solution for G.[1] The following result is known regarding Weighted Ptolemaic Deletion.

▶ **Proposition 4** (Theorem 1.1 [7]). *There is a constant $c \in \mathbb{N}$ and a polynomial time algorithm* Approx *for* Weighted Ptolemaic Deletion*, which given a graph G and a weight function $w : V(G) \to \mathbb{R}^+ \cup \{0\}$, outputs a solution for G of weight at most $c \cdot \mathsf{opt}(G)$.*

2.1 Computing a Redundant Solution

Our kernelization algorithm will also use a "strengthened" approximate solution, which will be slightly larger in size than the solution that we obtain using the known constant-factor approximation algorithm (Proposition 4). The type of strengthening of an approximate solution that we use will be the same as the notion of *redundant solution* that was introduced in the context of Interval Vertex Deletion [4]. Intuitively speaking, a redundant solution allows us to "forget" about "small" obstructions in some of our reduction rules. This is achieved with the help of an implicit family of subsets of $V(G)$ which has a guarantee that any solution for G of size at most k is a hitting set for this family. We will now formalise the above notions.

Consider a graph G. For a family $\mathcal{W} \subseteq 2^{V(G)}$, a subset $S \subseteq V(G)$ *hits* $\mathcal{W}$ if for all $W \in \mathcal{W}$, we have $S \cap W \neq \emptyset$. We say that $\mathcal{W} \subseteq 2^{V(G)}$ is *t-necessary* if every solution for G 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 the notion of redundant solution (for our context).

▶ **Definition 5.** For a graph G, 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 any obstruction $\mathbb{O}$ that is not covered by $\mathcal{W}$, it holds that $|M \cap V(\mathbb{O})| > t$.

Let (G, k) be an instance of Ptolemaic Deletion. We will next focus on computing a t-redundant solution for G, for an appropriately defined $\mathcal{W}$ (and t).

▶ **Lemma 6.** [♠] [2] *In polynomial time we can either correctly conclude that (G, k) is a* no*-instance, or compute a $(k+1)$-necessary family $\mathcal{W} \subseteq 2^{V(G)}$ and a set $M \subseteq V(G)$, such that $\mathcal{W} \subseteq 2^M$, M is a solution for G that is 4-redundant with respect to $\mathcal{W}$ and $|M| \in O(k^5)$.*

The utility of a redundant solution is captured in the following lemma.

▶ **Lemma 7.** [♠] *Consider an instance (G, k) of* Ptolemaic Deletion*, and D be a 4-redundant solution for G with respect to a $(k+1)$-necessary family $\mathcal{W}$, returned by Lemma 6. For $v \in V(G) \setminus D$, let X be any solution of size at most k (if it exists) for $G \setminus \{v\}$. Then X hits all obstructions of size at most 5 in G. In other words, $G \setminus X$ does not contain a gem or a chordless cycle of length less than 6.*

[1] For a subset $S \subseteq V(G)$, $w(S) = \sum_{s \in S} w(S)$.
[2] Proofs of results marked with ♠ can be found in the full version of the paper.

3 Kernel for Ptolemaic Vertex Deletion

The objective of this section is to prove Theorem 1. Let (G, k) be an instance of Ptolemaic Deletion. Our algorithm begins by computing a c-approximate solution S, for the Weighted Ptolemaic Deletion instance $(G, w^*_{n+1,\emptyset})$ using Approx (see Proposition 4).[3] If $|S| > ck$, we conclude that (G, k) is a no-instance, and return a trivial no-instance of the problem as a kernel. We now assume $|S| \leqslant ck$. Next, we use Lemma 6 to compute (in polynomial time) a $(k+1)$-necessary family $\mathcal{W}$ and a 4-redundant (with respect to $\mathcal{W}$) solution D for G of size bounded by $\mathcal{O}(k^5)$. Note that we have $\mathcal{W} \subseteq 2^D$, from the lemma.

Firstly, we bound the size of a maximal clique in $G \setminus (S \cup D)$ using the structure of the connected components upon removal of a maximal clique in a Ptolemaic graph and the properties guaranteed by the 4-redundant solution D. Intuitively, for each maximal clique C in $G \setminus S$, we do the following. Firstly we mark a few vertices in C which are neighbours of some vertices in S. Secondly, we create auxiliary graphs, with the guarantee that: i) if the maximum matching in these auxiliary graphs is "large", we find a vertex v that can be safely deleted to obtain the instance $(G \setminus \{v\}, k-1)$, and ii) otherwise, we will be able to use maximum matchings in these auxiliary graphs to mark at most $\mathcal{O}(k^3)$ many vertices in C. The marked neighbours of S in C and the vertices marked using the help of auxiliary graphs will help us capture different behaviours of C in an obstruction. Finally we argue that deleting unmarked vertices in $C \setminus D$ is safe. Formally, we prove the following lemma in Section 4.

▶ **Lemma 8.** *In polynomial time we can either conclude that (G, k) is a* no*-instance, or obtain an equivalent instance (G', k'), such that $k' \leqslant k$, G' is an induced subgraph of G, such that $S \cup D \subseteq V(G')$, and each (maximal) clique in $G' \setminus (S \cup D)$ has at most $\mathcal{O}(k^3)$ vertices.*

If at any point in our algorithm, we are able to conclude that the given instance is a no-instance, then we output some trivial no-instance as a kernel with $\mathcal{O}(1)$ vertices. We use Lemma 8 for the given instance, and without loss of generality assume that we obtained an equivalent instance of Ptolemaic Deletion, which satisfies all the conditions guaranteed by the lemma. For the sake of notational simplicity, we use (G, k) as the instance of Ptolemaic Deletion that we currently have, i.e., the one returned by the lemma.

Next, we use the fact that the undirected version of the inter-clique digraph of a Ptolemaic graph is an undirected forest. We construct a suitable Ptolemaic graph $G \setminus (S \cup R)$ for a carefully constructed set $R \subseteq D$ and reduce the number of leaves in its undirected inter-clique digraph. Towards this, we use the idea of bounding the independence degree of vertices from [3]. We introduce two sets of pairs of vertices E_M and E_I, which will stand for mandatory and irrelevant pairs of vertices. Moreover, we call an edge $\{u, v\} \in E(G)$ *mandatory* if $\{u, v\} \in E_M$ and *irrelevant* if $\{u, v\} \in E_I$. An edge is *relevant* if it is not irrelevant.

For a graph H and sets $E_M, E_I \subseteq V(H) \times V(H)$, for a vertex $v \in V(H)$, by $N^H_{\mathsf{rel}}(v)$ we denote the set of vertices in the neighbourhood of v in H which are adjacent to v using an edge in $E(H) \setminus E_I$. We omit the superscript H from the above notation, whenever the context is clear.

▶ **Definition 9.** *Given a set $E_I \subseteq V(G) \times V(G)$ and a vertex $v \in V(G)$, the independence degree of v in G, denoted by $d^G_I(v)$, is the size of a maximum independent set in the graph $G[N_{\mathsf{rel}}(v)]$.*

[3] The choice of $n+1$ is arbitrary.

We remark that we omit the superscript in $d_I^G(v)$ when the graph is clear from the context. Next, we consider an annotated version of Ptolemaic Deletion similar to [3].

> Augmented Ptolemaic Deletion **Parameter:** k
> **Input:** An undirected graph G, sets $E_M, E_I \subseteq V(G) \times V(G)$ and a non-negative integer k, such that any $X \subseteq V(G)$ which hits all chordless cycles in G which contain no edge from E_I, hits all chordless cycles in G.
> **Question:** Does there exist a subset $X' \subseteq V(G)$ (called a *solution* for (G, k, E_M, E_I)), such that (i) $|X'| \leqslant k$ (ii) For each $\{u, v\} \in E_M$, $X' \cap \{u, v\} \neq \emptyset$ and (iii) $G \setminus X'$ is a Ptolemaic graph?

We next obtain an instance of Augmented Ptolemaic Deletion, where the number of leaves in the inter-clique digraph of the Ptolemaic graph obtained after removing S and a suitable subset of D is bounded by $\mathcal{O}(k^3)$.

▶ **Lemma 10.** [♠] *In polynomial time, either we can conclude that (G, k) is a* no*-instance, or output an instance (G', k', E'_M, E'_I) of* Augmented Ptolemaic Deletion*, a set $R \subseteq D$ and the inter-clique digraph $T_{G' \setminus (R \cup S)}$ of $G' \setminus (R \cup S)$ so that the following hold:*

1. *$k' \leqslant k$, $S, D \subseteq V(G')$, and G' is an induced subgraph of G.*
2. *for each $\{u, v\} \in E'_M$, either $u \in S$ or $v \in S$.*
3. *Each chordless cycle in G' which contains a vertex from R contains an edge from E'_I.*
4. *(G, k) is a* yes*-instance of* Ptolemaic Deletion *if and only if (G', k', E'_M, E'_I) is a* yes*-instance of* Augmented Ptolemaic Deletion*.*
5. *The neighbourhood of every vertex $r \in R$ in $G \setminus (R \cup S)$ is a clique.*
6. *The number of leaves in $T_{G' \setminus (R \cup S)}$ is bounded by $\mathcal{O}(k^3)$.*
7. *The independence degree (in G') of every vertex $s \in S$ is bounded by $\mathcal{O}(k^2)$ and $|E'_M| \in \mathcal{O}(k^2)$.*

We use Lemma 10, and assume that we have an instance (G', k', E'_M, E'_I) of Augmented Ptolemaic Deletion and a set $R \subseteq D$, satisfying the properties guaranteed by the lemma. Using the next lemma we obtain an instance of Augmented Ptolemaic Deletion equivalent to that of (G', k', E'_M, E'_I), with $\mathcal{O}(k^6)$ vertices.

▶ **Lemma 11.** [♠] *For the* Ptolemaic Deletion *instance (G, k), let (G', k', E'_M, E'_I) be the instance of* Augmented Ptolemaic Deletion *and $R \subseteq D$ be the set returned by Lemma 10.*[4] *In polynomial time we either conclude that (G', k', E'_M, E'_I) is a* no*-instance or output an equivalent instance $(\widetilde{G}, \widetilde{k}, \widetilde{E}_M, \widetilde{E}_I)$ of* Augmented Ptolemaic Deletion*, such that $\widetilde{k} \leqslant k'$, $|V(\widetilde{G})| \in \mathcal{O}(k^6)$, and $|\widetilde{E}_M| \leqslant |E'_M|$.*

To prove the above lemma, we reduce the number of vertices in 2-degree bags in the undirected inter-clique digraph of $G' \setminus (R \cup S)$. We do this by exploiting the structure of 2-degree paths, which are paths in $T_{G' \setminus (R \cup S)}$ where each bag is of degree 2. On a high level, we mark many bags in $T_{G' \setminus (R \cup S)}$ so that each 2-degree path splits into several maximal unmarked sub-paths. We ensure that the vertices of G' in bags of these subpaths do not "interact" with $R \cup S$, that is, they do not have any neighbours in $R \cup S$, using several reduction rules. We then show that we can reduce the number of vertices in these bags, by designing a reduction rule which preserves the size of a minimum separator in an augmented graph. Since we already have a bound on the number of leaves from Lemma 11, we get a

[4] We assume that in G, each maximal clique has at most $\mathcal{O}(k^3)$ vertices, using Lemma 8.

bound on the number of degree $\geqslant 3$ bags as well. Thus, proving a bound on the number of vertices present in bags of degree 2 gives us a bound on the total number of vertices in the inter-clique digraph of $G' \setminus (R \cup S)$, since we already have a bound on clique size (and hence, on the number of vertices in any bag) from Lemma 8. Accounting for each step we obtain an Augmented Ptolemaic Deletion instance with $\mathcal{O}(k^6)$ vertices.

Equipped with Lemmas 8 to 11, we are now ready to prove Theorem 1.

Proof of Theorem 1. Consider an instance $(\widehat{G}, \widehat{k})$ of Ptolemaic Deletion. If Lemma 8 returns that $(\widehat{G}, \widehat{k})$ is a no-instance of Ptolemaic Deletion, then we return some trivial no-instance with $\mathcal{O}(1)$ vertices as a kernel for Ptolemaic Deletion. Otherwise, we have an equivalent instance (G, k), where each (maximal) clique in $G \setminus (S \cup D)$ has at most $\mathcal{O}(k^3)$ vertices. Next we apply Lemma 10 for the instance (G, k). Again, if the lemma returns that (G, k) is a no-instance, we return a trivial $\mathcal{O}(1)$-vertex kernel. Otherwise, we have an Augmented Ptolemaic Deletion instance (G', k', E'_M, E'_I) and the set $R \subseteq D$, returned by the lemma. We next apply Lemma 11. Firstly consider the case when the lemma returns that (G', k', E'_M, E'_I) is a no-instance of Augmented Ptolemaic Deletion. Then together with Lemma 10, we have that (G, k), and hence, $(\widehat{G}, \widehat{k})$ is a no-instance of Ptolemaic Deletion. Now consider the case when Lemma 11 returns an instance $(\widetilde{G}, \widetilde{k}, \widetilde{E}_M, \widetilde{E}_I)$ of Augmented Ptolemaic Deletion, where $|V(\widetilde{G})| \in \mathcal{O}(k^6)$. From Lemmas 8 to 11, we obtain that $(\widehat{G}, \widehat{k})$ is a yes-instance of Ptolemaic Deletion if and only if $(\widetilde{G}, \widetilde{k}, \widetilde{E}_M, \widetilde{E}_I)$ is a yes-instance of Augmented Ptolemaic Deletion.

We next obtain an instance (G', k') of Ptolemaic Deletion such that $|V(G')| \in \mathcal{O}(k^6)$ and (G', k') is a yes-instance of Ptolemaic Deletion if and only if $(\widetilde{G}, \widetilde{k}, \widetilde{E}_M, \widetilde{E}_I)$ is a yes-instance of Augmented Ptolemaic Deletion. Initialize $G' = \widetilde{G}$ and set $k' = \widetilde{k}$. For each $\{u, v\} \in \widetilde{E}_M$, add $k + 2$ vertex disjoint paths between u and v using 2 new vertices for each path. As $|\widetilde{E}_M| \leqslant |E'_M| \in \mathcal{O}(k^2)$, we obtain that $|V(G')| \in \mathcal{O}(k^6)$. Notice that no gem in G' can contain a vertex from $V(G') \setminus V(\widetilde{G})$. For every $\{u, v\} \in \widetilde{E}_M$ and any solution X to the Ptolemaic Deletion instance (G', k') , we must have either $u \in X$ or $v \in X$, for if not, then $G' \setminus X$ must contains a chordless cycle through u, v by construction. It follows that (G', k') is a yes-instance of Ptolemaic Deletion if and only if $(\widetilde{G}, \widetilde{k}, \widetilde{E}_M, \widetilde{E}_I)$ is a yes-instance of Augmented Ptolemaic Deletion. Hence, it must be the case that (G', k') and $(\widehat{G}, \widehat{k})$ are equivalent instances of Ptolemaic Deletion, where $|V(G')| \in \mathcal{O}(k^6)$ and $k' \leqslant k$. This concludes the proof. ◀

Since the proof of each of the three main lemmas is quite involved, in this short version we defer the proofs of Lemma 10 and Lemma 11 to the full version in the interest of space. At the same time we try to present all the results towards proving Lemma 8, to the extent possible.

4 Bounding Sizes of Maximal Cliques in $G \setminus (S \cup D)$

The objective of this section is to prove Lemma 8. As S is a solution for G, $G \setminus S$ does not have any chordless cycles. We next state a well-known result regarding enumerating maximal cliques in chordal graphs.

▶ **Proposition 12** (see, for example, Theorem 4.8 [14]). *In polynomial time we can compute the set of all maximal cliques in a given chordal graph.*

From the above proposition, the number of distinct cliques and the time required to enumerate them, are both bounded by $n^{\mathcal{O}(1)}$. Thus, for the remainder of this section, we work with a fixed maximal clique C in $G \setminus S$, and our objective will be to either conclude that the number of vertices in $C \setminus D$ is bounded by $\mathcal{O}(k^3)$, or find a vertex in C which we can safely delete from G. To achieve this, we will design some marking schemes, and show that if there are unmarked vertices, then we can safely delete them. We start with a useful lemma regarding the structure of connected components upon removal of a maximal clique from a Ptolemaic graph.

▶ **Lemma 13.** *Let $\widehat{G}$ be a Ptolemaic graph and C be a maximal clique in $\widehat{G}$. Let A be a connected component of $\widehat{G} \setminus C$. Then for every vertex $v \in V(A)$ we must have $N_{\widehat{G}}(v) \cap C = \emptyset$ or $N_{\widehat{G}}(v) \cap C = N_{\widehat{G}}(V(A)) \cap C$.*

Proof. Consider a connected component A of $\widehat{G} \setminus C$. Towards a contradiction suppose that the result is not true, and let $u, v \in V(A)$ be two distinct vertices such that $N_{\widehat{G}}(u) \cap C, N_{\widehat{G}}(v) \cap C \neq \emptyset$ and $N_{\widehat{G}}(u) \cap C \neq N_{\widehat{G}}(v) \cap C$. Furthermore, let u and v be a pair of vertices satisfying the above property that have shortest possible distance between them in A. Note that $C \not\subseteq N_{\widehat{G}}(u)$ and $C \not\subseteq N_{\widehat{G}}(v)$, as C is a maximal clique in $\widehat{G}$.

Suppose that $\{u, v\} \in E(\widehat{G})$. If there exist vertices $a, b \in V(C)$ such that $a \in N_{\widehat{G}}(u) \setminus N_{\widehat{G}}(v)$, $b \in N_{\widehat{G}}(v) \setminus N_{\widehat{G}}(u)$ then $\widehat{G}[\{u, v, a, b\}]$ is a chordless cycle in $\widehat{G}$, which is a contradiction, as $\widehat{G}$ is Ptolemaic graph. Therefore, either $N_{\widehat{G}}(u) \cap C \subseteq N_{\widehat{G}}(v) \cap C$ or $N_{\widehat{G}}(v) \cap C \subseteq N_{\widehat{G}}(u) \cap C$. Assume without loss of generality that $N_{\widehat{G}}(u) \cap C \subseteq N_{\widehat{G}}(v) \cap C$. This together with the choice of u and v implies that there are vertices $a \in (N_{\widehat{G}}(v) \cap C) \setminus N_{\widehat{G}}(u)$, $b \in N_{\widehat{G}}(u) \cap C$ and $x \in C \setminus (N_{\widehat{G}}(u) \cup N_{\widehat{G}}(v))$. But then, $\widehat{G}[\{a, b, x, u, v\}]$ is a gem in $\widehat{G}$, which is a contradiction.

Now suppose $\{u, v\} \notin E(\widehat{G})$ and consider the shortest path P between u, v in A. Note that P has at least one more vertex apart from u and v, as $\{u, v\} \notin E(\widehat{G})$. Firstly consider the case when there are vertices $a \in (N_{\widehat{G}}(u) \cap C) \setminus N_{\widehat{G}}(v)$, $b \in (N_{\widehat{G}}(v) \cap C) \setminus N_{\widehat{G}}(u)$. Note that by the choice of u and v (of them being at shortest distance in A satisfying the assumed properties), no internal vertex of P can be adjacent to a or b. But then P along with the edges $\{a, b\}, \{a, u\}$ and $\{v, b\}$ forms a chordless cycle in $\widehat{G}$, which is a contradiction. We now consider the case when there are no such a and b, as assumed previously. This implies that either $N_{\widehat{G}}(u) \cap C \subseteq N_{\widehat{G}}(v) \cap C$ or $N_{\widehat{G}}(v) \cap C \subseteq N_{\widehat{G}}(u) \cap C$. Suppose that $N_{\widehat{G}}(u) \cap C \subseteq N_{\widehat{G}}(v) \cap C$ (the other case is symmetric). Now consider a vertex $a \in N_{\widehat{G}}(v) \cap N_{\widehat{G}}(u) \cap C$. If there is an internal vertex of P that is not adjacent to a, then we conclude that $\widehat{G}[V(P) \cup \{a\}]$ contains a chordless cycle. If every internal vertex of P is adjacent to a, P must have exactly 1 vertex, say w, apart from $\{u, v\}$, since otherwise we would obtain a gem obstruction in $\widehat{G}[V(P) \cup \{a\}]$. Since $N_{\widehat{G}}(u) \cap C \neq N_{\widehat{G}}(v) \cap C$ we must have either $N_{\widehat{G}}(w) \cap C \neq N_{\widehat{G}}(u) \cap C$ or $N_{\widehat{G}}(w) \cap C \neq N_{\widehat{G}}(v) \cap C$. In either case, since $\{u, w\}, \{w, v\} \in E(\widehat{G})$, this is impossible from the analysis in the preceeding paragraph of the case when $\{u, v\} \in E(\widehat{G})$. It follows that such an a cannot exist, and therefore $N_{\widehat{G}}(v) \cap N_{\widehat{G}}(u) \cap C = \emptyset$. Together, we must have either $N_{\widehat{G}}(v) \cap C = \emptyset$ or $N_{\widehat{G}}(u) \cap C = \emptyset$, a contradiction. ◀

Recall that we computed D, a 4-redundant solution with respect to $\mathcal{W}$. This will allow us to only focus on chordless cycles of length at least 6 (see Lemma 7). A chordless cycle can contain at most 2 vertices from C, as C is a (maximal) clique. If C has many vertices, we would like to argue that, we can find a vertex $c \in C \setminus D$ to delete from G. Roughly speaking, to achieve this, for every pair of vertices u, v, that are neighbours of c in some chordless cycle, we would like to ensure that, after removing some solution of size at most k, there is either (i) at least one marked common neighbour $c' \in C$ of u and v or (ii) two distinct marked vertices $c_1, c_2 \in C$, such that $\{u, c_1\}, \{v, c_2\} \in E(G)$. We show, through a simple observation, that preserving such marked vertices is enough to safely delete the vertex c from the input instance (G, k) to obtain the instance $(G \setminus \{c\}, k)$.

▶ **Observation 14.** *Consider a graph $\widehat{G}$ and a spanning cycle K of $\widehat{G}$. Then the graph $\widehat{G}$ contains an obstruction if one of the following holds: i) $|V(\widehat{G})| \geqslant 5$ and there is $v \in V(K)$, such that for each $\{x,y\} \in E(\widehat{G}) \setminus E(K)$, $v \in \{x,y\}$, or ii) $|V(\widehat{G})| \geqslant 7$ and there is $\{u,v\} \in E(K)$, such that for each $\{x,y\} \in E(\widehat{G}) \setminus E(K)$, we have $\{u,v\} \cap \{x,y\} \neq \emptyset$.*

Proof. To prove (i), notice that if v is not adjacent (in $\widehat{G}$) to at least one vertex in $V(K)\setminus\{v\}$, then $\widehat{G}$ must contain a chordless cycle. Otherwise, since K has at least 5 vertices, any four consecutive vertices a, b, c, d on K, all different from v, together with v give us a gem.

To prove (ii), let us consider five consecutive vertices of the cycle K starting from u, v, taken in order, say u, v, a, b, c. We claim that apart from its two neighbours u and a in K, the vertex v can be adjacent to only the vertices b and c in $\widehat{G}$. Suppose this is not the case. Then we get a cycle K' with $V(K') \subseteq V(K)$ in $\widehat{G}$ of length at least 5 which does not include u. Applying part (i) with the graph $\widehat{G}[V(K')]$ then gives an obstruction in $\widehat{G}$.

Therefore v can only be adjacent to u, a, b, c in K. Consider the case when v is adjacent to b but not c. Let K' be the cycle in $\widehat{G}$, obtained from K by deleting a and adding the edge $\{v, b\}$. Notice that $\widehat{G}[V(K')]$ and the spanning cycle K' of $\widehat{G}[V(K')]$, satisfy part (i) of the observation. Thus, $\widehat{G}[V(K')]$ (and hence $\widehat{G}$) must contain an obstruction. For the case when v is adjacent to c, we can obtain an obstruction in $\widehat{G}$ by using arguments similar to our previous case. Finally, if v is not adjacent to both b and c, then $\widehat{G}$ and K satisfy the conditions of (i). ◀

Now we define the notions of replacement vertices.

▶ **Definition 15.** Given a vertex $c \in C$, a set $X \subseteq V(G) \setminus \{c\}$ of size at most k, and a chordless cycle K of length at least 6 in $G \setminus X$ containing c, a vertex $c' \in C \setminus (X \cup V(K))$ is a *replacement* for c in K, if c' is adjacent to both the neighbours of c in K.

▶ **Definition 16.** Given a vertex $c \in C$, a set $X \subseteq V(G) \setminus \{c\}$ of size at most k, and a chordless cycle K of length at least 6 in $G \setminus X$ containing c, a pair of distinct vertices $c_1, c_2 \in C \setminus (X \cup V(K))$, is a *replacement* for c in K, if $\{u, c_1\}, \{v, c_2\} \in E(G)$, where u, v are the neighbours of c in K.

The purpose of this notion of replacements for a vertex is captured in the following lemma whose proof follows from Observation 14.

▶ **Lemma 17.** *Consider any vertex $c \in C$. Let $X \subseteq V(G)$ be a set of vertices such that $c \notin X$. Suppose that $G \setminus X$ contains a chordless cycle K of length at least 6 which includes the vertex c. Now suppose that there is either a single replacement vertex c' or a pair of replacement vertices c_1, c_2 for c in K, such that $c' \in C \setminus (X \cup V(K))$ and $c_1, c_2 \in C \setminus (X \cup V(K))$, as the case may be. Then $G \setminus (X \cup \{c\})$ must contain an obstruction.*

Proof. Let u and v be the neighbours of c in K. If we have a single replacement vertex c' for c in K, then consider the cycle K' obtained by replacing the path ucv by the path $uc'v$ in K. The cycle K' is clearly of length at least 6. Applying part (i) of Observation 14 to the graph $G[V(K')]$, we must have an obstruction in $G \setminus (X \cup \{c\})$. Likewise, if we have a pair of distinct vertices $c_1, c_2 \in C$ as a replacement for c in K, then consider the cycle K' obtained by replacing the path ucv by the path uc_1c_2v. The cycle K' is of length at least 7, therefore by item (ii) of Observation 14 applied to the graph $G[V(K')]$, $G \setminus (X \cup \{c\})$ must contain an obstruction. ◀

Now suppose that we mark a certain number of vertices in C. Consider any unmarked vertex $c \in C \setminus D$, and let X be a solution of size at most k for $G \setminus \{c\}$. If X is not a solution for G, then $G \setminus X$ must contain an obstruction that contains the vertex c. This

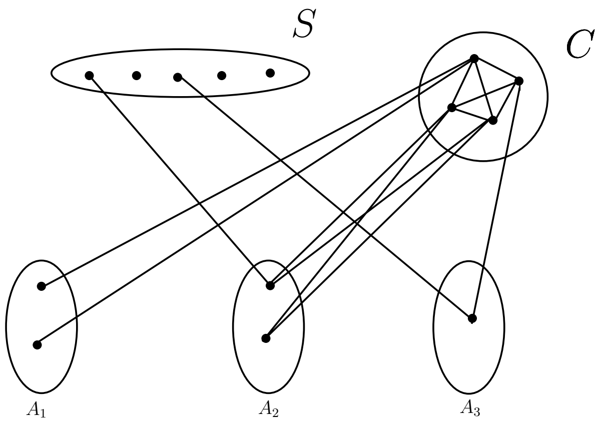

Figure 1 Bounding the size of a maximal clique C. The figure shows the components of $G \setminus (S \cup C)$ for a particular instance. For every component A and vertex $v_A \in A$, $N(v_A) \cap C = \emptyset$ or $N(v_A) \cap C = N(A) \cap C$.

obstruction must be a chordless cycle K of length at least 6 since $c \notin D$. If we design our marking scheme in such a way that for every such obstruction K, there is either (i) a marked replacement vertex c' such that $c' \notin (X \cup V(K))$ or (ii) a pair of distinct marked replacement vertices c_1, c_2 for c in K such that $c_1, c_2 \notin (X \cup V(K))$ as per the definitions above, then by Lemma 17, $G \setminus (X \cup \{c\})$ must contain an obstruction - a contradiction since X was a solution for $G \setminus \{c\}$. It follows that X must be a solution for G, and therefore the instances (G, k) and $(G \setminus \{c\}, k)$ are equivalent. Thus, given that we design our marking scheme to satisfy the above, we may delete an unmarked vertex from the input instance G without reducing the parameter.

We now describe our marking scheme and formalize the ideas from above, by looking at where the neighbours of an unmarked vertex in a chordless cycle can potentially lie. Whenever we say - mark any ℓ vertices satisfying a property P, if there are less than ℓ vertices satisfying P, then we mark all vertices which satisfy P. We start by designing our marking scheme to handle the case when the neighbours of an unmarked vertex in a chordless cycle lie in S.

▶ **Marking Scheme 1.** *For every pair of (not necessarily distinct) vertices $s_1, s_2 \in S$, mark any $k + 2$ vertices in $N(s_1) \cap N(s_2) \cap C$.*

To consider the cases when the neighbours of a potential unmarked vertex lie in connected components of $G \setminus (S \cup C)$, we make use of two auxiliary bipartite graphs H_s^1 and H_s^2, corresponding to each vertex s in S. Let us denote the set of connected components of $G \setminus (S \cup C)$ by $\mathcal{A}$. The vertex set of H_s^1 is $V_s^1 \cup V_2$ where $V_s^1 = \{c \in C \ : (s, c) \notin E(G)\}$ and $V_2 = \{v_A \ : A \in \mathcal{A}\}$. We add an edge from a vertex $c \in V_s^1$ and a vertex $v_A \in V_2$ whenever both c and s have (possibly different) neighbours in A. The vertex set of H_s^2 is $V_s^2 \cup V_2$, where $V_s^2 = \{c \in C \ : (s, c) \in E(G)\}$. We add an edge from a vertex $c \in V_s^2$ and a vertex $v_A \in V_2$ whenever there is an obstruction in the subgraph induced on $\{s, c\} \cup V(A)$. We have the following observation that follows from the above description and Proposition 3.[5]

▶ **Observation 18.** *For $s \in S$, H_s^1 and H_s^2 can be constructed in polynomial time.*

[5] We note that there are other well-known polynomial time recognition algorithm for Ptolemaic graphs, that can also be used to infer that H_s^2, for $s \in S$, can be constructed in polynomial time.

Roughly speaking, we next show that, for $s \in S$ and two edges of H_s^1, incident to different vertices in V_2, we can construct an obstruction in G.

▶ **Lemma 19.** *Consider $s \in S$ and edges $\{c_1, v_{A_1}\}, \{c_2, v_{A_2}\} \in E(H_s^1)$, where $c_1, c_2 \in V_s^1$ and $A_1, A_2 \in \mathcal{A}$, where $A_1 \neq A_2$. Then, $G[V(A_1) \cup V(A_2) \cup \{s, c_1, c_2\}]$ has a chordless cycle.*

Proof. By the construction of V_s^1, $\{s, c_1\}, \{s, c_2\} \notin E(G)$. For $i, j \in \{1, 2\}$, let P_{ij} (if it exists) be a shortest path between s and c_i such that every internal vertex is from A_j. By the construction of H_s^1, P_{11} and P_{22} must necessarily exist. Notice that if at least one of c_1 or c_2, say c_1, is a neighbour of both v_{A_1} and v_{A_2} in H_s^1, then $G[\{s, c_1\} \cup V(P_{11}) \cup V(P_{12})]$ is a chordless cycle. Otherwise, the vertices s, c_1, c_2 together with the paths P_{11}, P_{22} and the edge $\{c_1, c_2\}$ constitute a chordless cycle $G[\{s, c_1, c_2\} \cup V(P_{11}) \cup V(P_{22})]$ in $G[V(A_1) \cup V(A_2) \cup \{s, c_1, c_2\}]$. ◀

For $s \in S$, we now attempt to find $(k + 2)$-sized matchings in the graphs H_s^1 and H_s^2. Note that the existence of a $(k + 2)$-sized matching in H_{s1} would imply that (at least) two matching edges "remain", even after we delete at most k vertices from G. The above together with Lemma 19 will allow us to conclude that s must belong to every solution for G of size at most k. The definition of H_s^2 implies that even a single "remaining" matching edge in H_s^2 would give us an obstruction, if s is not included in the solution.

▶ **Lemma 20.** *Consider $s \in S$, and let M_s^1 and M_s^2 be maximum matchings in H_s^1 and H_s^2, respectively. If either $|M_s^1| \geqslant k + 2$ or $|M_s^2| \geqslant k + 2$, then any solution for G of size at most k must include the vertex s.*

Proof. Towards a contradiction, suppose that X is a solution for G of size at most k which does not include the vertex s. Consider the case when $|M_s^1| \geqslant k+2$. It follows that there are distinct components $A_1, A_2 \in \mathcal{A}$, and vertices $c_1, c_2 \in V_s^1$, so that (i) $X \cap (V(A_1) \cup V(A_2) \cup \{c_1, c_2\}) = \emptyset$ and (ii) $\{c_1, v_{A_1}\}, \{c_2, v_{A_2}\} \in E(H_s^1)$ hold. But then Lemma 19 implies that $G \setminus X$ contains an obstruction, which is a contradiction. Next consider the case when $|M_s^2| \geqslant k+2$. Again as X is of size at most k, we can obtain that there is a component $A \in \mathcal{A}$ and a vertex $c \in V_s^2$ so that (i) $X \cap (V(A) \cup \{c\}) = \emptyset$ and (ii) $\{c, v_A\} \in E(H_s^2)$ hold. By the definition of H_s^2, this implies that $G \setminus X$ contains an obstruction in the subgraph induced on $V(A) \cup \{s, c\}$, which is a contradiction. ◀

Using the above lemma, we devise our first reduction rule.

▶ **Reduction Rule 1.** *If there is $s \in S$, such that $|M_s^1| \geqslant k + 2$ or $|M_s^2| \geqslant k + 2$, then return $(G \setminus \{s\}, k - 1)$.*

The correctness of the above reduction rule follows from Lemma 20, and the polynomial time applicability of it follows from the fact that maximum matching in a graph can be computed in polynomial time (see, for example [12]). Hereafter we assume that Reduction rule 1 is not applicable, and we proceed with our next marking scheme.

▶ **Marking Scheme 2.** *For each $s \in S$, let M_s^1 and M_s^2 be maximum matchings in H_s^1 H_s^2, respectively. Then we do the following: i) for each $i \in \{1, 2\}$, we mark all the vertices in $V(M_s^i) \cap C$, ii) for each $A \in \mathcal{A}$, where $v_A \in V(M_s^1)$, mark any $k+2$ vertices in $N(V(A)) \cap V_s^1$, and iii) for each $A \in \mathcal{A}$, where $v_A \in V(M_s^2)$, mark any $k + 2$ vertices in $N(V(A)) \cap V_s^2$.*

The next reduction rule deletes unmarked vertices in C.

▶ **Reduction Rule 2.** *If there is $c \in C \setminus D$ that is not marked by Marking Scheme 1 or 2, then return $(G \setminus \{c\}, k)$.*

As Marking Scheme 1 and 2 can be executed in polynomial time, the above reduction rule can be executed in polynomial time as well. In the next lemma we show its safeness.

▶ **Lemma 21.** *Reduction rule 2 is safe.*

Proof. Consider $c \in C \setminus D$ that is not marked by Marking Scheme 1 or 2. As Ptolemaic graphs are closed under induced subgraphs, to prove the lemma, it is enough to argue that a solution X for $G \setminus \{c\}$ of size at most k is also a solution for G. Towards a contradiction suppose that X is not a solution for G. Thus, $G \setminus X$ has an obstruction K, containing the vertex c. Since $c \notin D$, by Lemma 7, we may assume that K is a chordless cycle of length at least 6. Next we will construct an obstruction in $(G \setminus \{c\}) \setminus X$, and thus contradict that X is a solution for $G \setminus \{c\}$.

Firstly, consider the case when K has no other vertex from C, apart from c. Let u and v be the neighbours of c in K. Lemma 17 implies that, if we manage to find a replacement for c from $C \setminus (X \cup V(K))$ (see Definition 15 and 16), then we can find an obstruction in $(G \setminus \{c\}) \setminus X$. Firstly consider the case when $u, v \in S$. Since $k + 2$ common neighbours of u, v in C were marked by Marking Scheme 1, there is at least one common neighbour $c' \in (V(G) \setminus (X \cup \{c\})) \cap C$ which forms a replacement for c. This together with Lemma 17 and the premise of the case that K has no other vertex apart from c from C implies that $(G \setminus \{c\}) \setminus X$ contain an obstruction.

Recall that we are in the case when K contain exactly one vertex, namely, c from C. Next we suppose that $u \in S$ and $v \in V(A)$, for some $A \in \mathcal{A}$. Let s be the first vertex from S on the path from v to u in $K \setminus \{c\}$ (Clearly, such an s must exist since S is an approximate solution). Notice that every internal vertex on the path from c to s, containing v, on K must be from A. Consider the case when $u \neq s$. As K is a chordless cycle, we can obtain that $\{s, c\} \notin E(G)$, and thus $\{c, v_A\} \in E(H^1_s)$. Since c is unmarked, $c \notin V(M^1_s)$. Moreover, as M^1_s is a maximum matching in H^1_s, it follows that $v_A \in V(M^1_s)$. Recall that Marking Scheme 2 marked $k + 2$ vertices from $N(V(A)) \cap V^1_s$. Thus, there is a vertex $c' \in (C \setminus (X \cup \{c\})) \cap N(V(A)) \cap V^1_s$. By Lemma 13, since $\{v, c\} \in E(G)$ we must have $\{v, c'\} \in E(G)$. Marking Scheme 1 marked $k + 2$ neighbors of $u \in S$ from C. Thus, there must exist a vertex $c_u \in C \setminus (X \cup \{c\})$, such that $\{c_u, u\} \in E(G)$. If $c' = c_u$, the vertex c' is a replacement for c with respect to K, and otherwise, the pair of vertices c', c_u forms a replacement for c in K. In either case, using Lemma 17 and the fact that K is a chordless cycle on at least 6 vertices, we can obtain an obstruction in $(G \setminus \{c\}) \setminus X$, for the case when $u \neq s$. Next we consider the case when $u = s$. Notice that $\{c, v_A\}$ is an edge in H^2_s. As c is unmarked, it must be the case that $v_A \in V(M^2_s)$. Since $k+2$ vertices in $N(V(A)) \cap C \cap N(s)$ were marked by Marking Scheme 2, there is a vertex $c' \in (N(V(A)) \cap N(s) \cap C) \setminus (X \cup \{c\})$. Again, by Lemma 13 we obtain that $\{v, c'\} \in E(G)$, and therefore the vertex c' forms a replacement for c, and hence from Lemma 17 we can construct an obstruction in $(G \setminus \{c\}) \setminus X$.

Next we consider the case when $u \in V(A_1)$ and $v \in V(A_2)$, for some (not necessarily distinct) $A_1, A_2 \in \mathcal{A}$, and K has no other vertex from C, apart from c. Let s_1, s_2 be the first vertices from S on the paths from u to v and v to u in $K \setminus \{c\}$, respectively. Observe that $\{c, v_{A_1}\} \in E(H^1_{s_1})$ and $\{c, v_{A_2}\} \in E(H^1_{s_2})$. Since c is unmarked, we must have $v_{A_1} \in M^1_{s_1}$ and $v_{A_2} \in M^1_{s_2}$. Marking Scheme 2 ensures that there are c'_1, $c'_2 \in C \cap V(G) \setminus (X \cup \{c\})$, where $c'_1 \in N(V(A_1))$ and $c'_2 \in N(V(A_2))$ which, by Lemma 13, are adjacent to vertices u, v respectively. If $c'_1 = c'_2$, we use c'_1 as a replacement for c in K, else the pair c'_1, c'_2 forms a replacement for c.

Now we consider the case when the chordless cycle K contains two vertices from C. (Note that since K is a chordless cycle, it can contain at most 2 vertices from C.) Since K contains c, exactly one of $u, v \notin C$, say $u \notin C$ (the other case can be argued symmetrically). We further consider the following cases based on whether $u \in S$. Firstly consider the case when $u \in S$. Since Marking Scheme 1 marked $k+2$ neighbours of $u \in S$ from C, it follows that there is a vertex $c' \in C \setminus (X \cup V(K))$ which is a replacement for c, and thus using Lemma 17 we can obtain an obstruction in $(G \setminus \{c\}) \setminus X$. Now suppose that $u \in V(A)$ for some $A \in \mathcal{A}$. Let s be the first vertex from S on the path from u to v in $K \setminus \{c\}$. This implies that $\{c, v_A\} \in E(H_s^1)$ and that $v_A \in V(M_s^1)$. Therefore, $k+2$ vertices in $N((V(A)) \cap V_s^1$ must have been marked by Marking Scheme 2. Since $|X| \leqslant k$, c is an unmarked vertex, and $|C \cap V(K)| \leqslant 2$, there must exist a marked vertex $c' \in (N(V(A)) \cap V_s^1) \setminus (X \cup V(K))$. By Lemma 13 we must have $\{u, c'\} \in E(G)$. Thus, c' forms a replacement for c in K, and we obtain an obstruction in $(G \setminus \{c\}) \setminus X$, using Lemma 17. ◀

Proof of Lemma 8. For the given instance (G, k) of PTOLEMAIC DELETION, an approximate solution S for G of size $\mathcal{O}(k)$, and a solution D for G that is 4-redundant with respect to a $(k+1)$-necessary family $\mathcal{W}$, returned by Lemma 6, we do the following. In polynomial time, we enumerate the set $\mathcal{C}$, of all maximal maximal cliques in $G \setminus S$ using Observation 12. For each clique $C \in \mathcal{C}$, we exhaustively apply Reduction Rule 1 and 2, where the lowest applicable reduction rule is applied first. Note that each of these reduction rules can be executed in polynomial time, and they can only be applied polynomially many times. If none of the reduction rules are applicable, then we argue below that the number of vertices in $C \setminus D$ can be bounded by $\mathcal{O}(k^3)$. Note that Marking Scheme 1 marks at most $\mathcal{O}(k^3)$ vertices in C. Note that while executing Marking Scheme 2, Reduction Rule 1 is not applicable. Thus the number of vertices marked by Marking Scheme 2 are bounded by $\mathcal{O}(k^3)$ as for every $s \in S$, there can be at most $k+1$ matching edges in both the matchings M_s^1 and M_s^2, and corresponding to each edge we mark $\mathcal{O}(k)$ vertices in C. As the unmarked vertices of $C \setminus D$ are deleted by Reduction Rule 2, we must have $|C \setminus D| \in \mathcal{O}(k^3)$. ◀

References

1 Akanksha Agrawal, Sushmita Gupta, Pallavi Jain, and R. Krithika. Quadratic vertex kernel for split vertex deletion. *Theoretical Computer Science*, 833:164–172, 2020.

2 Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshtanov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In *Theoretical Informatics - 12th Latin American Symposium (LATIN)*, pages 1–13, 2016.

3 Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Trans. Algorithms*, 15(1), December 2018. doi:10.1145/3284356.

4 Akanksha Agrawal, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Interval vertex deletion admits a polynomial kernel. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 1711–1730, USA, 2019. Society for Industrial and Applied Mathematics.

5 Jungho Ahn. A polynomial kernel for 3-leaf power deletion. *Master's Thesis, KAIST, South Korea*, 2020.

6 Jungho Ahn, E. Eiben, O-joung Kwon, and Sang il Oum. A polynomial kernel for 3-leaf power deletion. In *MFCS*, 2020.

7 Jungho Ahn, Eun Jung Kim, and Euiwoong Lee. Towards constant-factor approximation for chordal/distance-hereditary vertex deletion. In *31st International Symposium on Algorithms and Computation (ISAAC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

8 Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms*, 11(3):21:1–21:35, 2015.

9 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer-Verlag, 2015.

10 Marek Cygan and Marcin Pilipczuk. Split vertex deletion meets vertex cover: New fixed-parameter and exact exponential-time algorithms. *Information Processing Letters*, 113(5):179–182, 2013. `doi:10.1016/j.ipl.2013.01.001`.

11 Michael Dom, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Error compensation in leaf power problems. *Algorithmica*, 44:363–381, April 2006. `doi:10.1007/s00453-005-1180-z`.

12 Jack Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

13 Eduard Eiben, Robert Ganian, and O-joung Kwon. A single-exponential fixed-parameter algorithm for distance-hereditary vertex deletion. *Journal of Computer and System Sciences*, 97:121–146, 2018.

14 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Elsevier, 2004.

15 Edward Howorka. A characterization of ptolemaic graphs. *Journal of Graph Theory*, 5(3):323–331, 1981.

16 Bart M. P. Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. *SIAM Journal on Discrete Mathematics*, 32(3):2258–2301, 2018.

17 Eun Jung Kim and O-joung Kwon. A polynomial kernel for distance-hereditary vertex deletion. In *Algorithms and Data Structures - 15th International Symposium (WADS)*, volume 10389, pages 509–520, 2017.

18 Jon M. Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley, 2006.

19 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

20 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

21 Ryuhei Uehara and Yushi Uno. Laminar structure of ptolemaic graphs with applications. *Discrete Appl. Math.*, 157(7):1533–1543, April 2009. `doi:10.1016/j.dam.2008.09.006`.

22 Mihalis Yannakakis. Node- and edge-deletion np-complete problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–264, 1978.