Towards Constant-Factor Approximation for Chordal / Distance-Hereditary Vertex Deletion

Jungho Ahn
Department of Mathematical Sciences, KAIST, Daejeon, South Korea
Discrete Mathematics Group, Institute for Basic Science (IBS), Daejeon, South Korea
junghoahn@kaist.ac.kr

Eun Jung Kim
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France
eun-jung.kim@dauphine.fr

Euiwoong Lee
Department of Computer Science, New York University, NY, USA
euiwoong@cs.nyu.edu

Abstract

For a family of graphs $\mathcal{F}$, Weighted $\mathcal{F}$-Deletion is the problem for which the input is a vertex weighted graph $G = (V, E)$ and the goal is to delete $S \subseteq V$ with minimum weight such that $G \backslash S \in \mathcal{F}$. Designing a constant-factor approximation algorithm for large subclasses of perfect graphs has been an interesting research direction. Block graphs, 3-leaf power graphs, and interval graphs are known to admit constant-factor approximation algorithms, but the question is open for chordal graphs and distance-hereditary graphs.

In this paper, we add one more class to this list by presenting a constant-factor approximation algorithm when $\mathcal{F}$ is the intersection of chordal graphs and distance-hereditary graphs. They are known as *ptolemaic graphs* and form a superset of both block graphs and 3-leaf power graphs above. Our proof presents new properties and algorithmic results on *inter-clique* digraphs as well as an approximation algorithm for a variant of Feedback Vertex Set that exploits this relationship (named Feedback Vertex Set with Precedence Constraints), each of which may be of independent interest.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases ptolemaic, approximation algorithm, linear programming, feedback vertex set

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.62

Related Version A full version of this paper is available at `https://arxiv.org/abs/2009.00809`.

Funding *Jungho Ahn*: the Institute for Basic Science (IBS-R029-C1). *Eun Jung Kim*: ANR JCJC project "ASSK" (ANR-18-CE40-0025-01). *Euiwong Lee*: Simons Collaboration on Algorithms and Geometry.

Acknowledgements A part of this research was done during the "2019 IBS Summer Research Program on Algorithms and Complexity in Discrete Structures", hosted by the IBS Discrete Mathematics Group.

1 Introduction

Given a family of graphs $\mathcal{F}$, we consider the following problem.

Weighted $\mathcal{F}$-Deletion
Input : A graph $G = (V, E)$ with vertex weights $w : V \to \mathbb{R}^+ \cup \{0\}$.
Question : Find a set $S \subseteq V$ of minimum weight such that $G \backslash S \in \mathcal{F}$.

© Jungho Ahn, Eun Jung Kim, and Euiwoong Lee;
licensed under Creative Commons License CC-BY
31st International Symposium on Algorithms and Computation (ISAAC 2020).
Editors: Yixin Cao, Siu-Wing Cheng, and Minming Li; Article No. 62; pp. 62:1–62:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

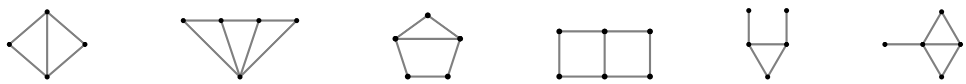

Figure 1 A diamond, a gem, a house, a domino, a bull, and a dart.

This problem captures many classical combinatorial optimization problems including Vertex Cover, Feedback Vertex Set, Odd Cycle Transversal, and the problems corresponding to natural graph classes (e.g., planar graphs, chordal graphs, or graphs of bounded treewidth) also have been actively studied. Most of these problems, including the simplest Vertex Cover, are NP-hard, so polynomial-time exact algorithms are unlikely to exist for them.

Parameterized algorithms and approximation algorithms have been two of the most popular kinds of algorithms for NP-hard optimization problems, and $\mathcal{F}$-Deletion has been actively studied from both viewpoints. There is a large body of work in the theory of parameterized complexity, where $\mathcal{F}$-Deletion for many $\mathcal{F}$'s is shown to be in FPT or even admits a polynomial kernel. The list of such $\mathcal{F}$'s includes chordal graphs [24, 17, 3], interval graphs [8, 7, 4], distance-hereditary graphs [11, 19], bipartite graphs [26, 22], and graphs with bounded treewidth [14, 21].

On the other hand, despite large interest, approximability for $\mathcal{F}$-Deletion is not as well as understood as parameterized complexity. To the best of our knowledge, for all $\mathcal{F}$'s admitting parameterized algorithms in the above paragraph except Odd Cycle Transversal, the existence of a constant-factor approximation algorithm is not ruled out under any complexity hypothesis. When $\mathcal{F}$ can be characterized by a finite list of forbidden subgraphs or induced subgraphs (not minors), the problem becomes a special case of Hypergraph Vertex Cover with bounded hyperedge size, which admits a constant-factor approximation algorithm. Besides them, the only classes of graphs that currently admit constant-factor approximation algorithms are block graphs [1], 3-leaf power graphs [5], interval graphs [7], and graphs of bounded treewidth [14, 15]. Weighted versions are sometimes harder than their unweighted counterparts, and within graphs of bounded treewidth, the only two nontrivial classes whose weighted version admits a constant-factor approximation algorithm are the set of forests (Weighted Feedback Vertex Set) and the set of graphs excluding a diamond as a minor [13]. See Figure 1.

When $\mathcal{F}$ is the set of perfect or weakly chordal graphs, it is known that a constant-factor approximation algorithm is unlikely to exist [16]. Therefore, there has been recent interest on identifying large subclasses of perfect graphs that admit constant-factor approximation algorithms. Among the subclasses of perfect graphs, chordal graphs and distance-hereditary graphs have drawn particular interest. Recall that chordal graphs are the graphs without any induced $C_{\geqslant 4}$[1], and distance-hereditary graphs are the graphs without any induced $C_{\geqslant 5}$, a gem, a house, or a domino. See Figure 1.

Chordal graphs are arguably the simplest graph class, apart from forests, which is characterized by infinite forbidden induced subgraphs. Structural and algorithmic aspects of chordal graphs have been extensively studied in the last decades, and it is considered one of the basic graph classes whose properties are well understood and on which otherwise NP-hard problems become tractable. As such, it is natural to ask how close a graph to a chordal graph in terms of graph edit distance and there is a large body of literature pursuing this topic [3, 2, 9, 17, 18, 24, 29].

[1] Let $C_{\geqslant k}$ be the set of cycles of length at least k.

Fixed-parameter tractability and the existence of polynomial kernel of $\mathcal{F}$-Deletion for chordal graphs were one of important open questions in parameterized complexity [24, 17]. An affirmative answer to the latter in [17] brought the approximability for chordal graphs to the fore as it uses an $O(\mathsf{opt}^2 \log \mathsf{opt} \log n)$-factor approximation algorithm as a crucial subroutine. It was soon improved to $O(\mathsf{opt} \log n)$-factor approximation [3, 20]. An important step was taken by Agrawal et al. [2] who studied Weighted $\mathcal{F}$-Deletion for chordal graphs, distance-hereditary graphs, and graphs of bounded treewidth. They presented polylog(n)-approximation algorithms for them, including $O(\log^2 n)$-approximation for chordal graphs, and left the existence of constant-factor approximation algorithms as an open question. For now, even the existence of $O(\log n)$-factor approximation is not known. This makes an interesting contrast with $\mathcal{F}$-Deletion for forests, that is, Feedback Vertex Set. An algorithmic proof of Erdös-Pósa property[2] for cycles immediately leads to an $O(\log n)$-factor approximation for Feedback Vertex Set while the known gap function of Erdös-Pósa property for induced $C_{\geqslant 4}$ is not low enough to achieve such an approximation factor [20].

Distance-hereditary graphs, in which any induced subgraph preserves the distances among all vertex pairs, form another important subclass of perfect graphs. It is supposedly the simplest dense graph class captured by a graph width parameter; distance-hereditary graphs are precisely the graphs of rankwidth 1 [25]. $\mathcal{F}$-Deletion for distance-hereditary graphs has gained good attention for fixed-parameter tractability and approximability [2, 19, 11] particularly due to the recent surge of interest in rankwidth. An $O(\log^3 n)$-approximation is known [2].

Constant-factor approximation algorithms were designed for smaller subclasses of chordal and distance-hereditary graphs. They include block graphs (excluding $C_{\geqslant 4}$ and a diamond) [1] and 3-leaf power graphs (excluding $C_{\geqslant 4}$, a bull, a dart, and a gem) [6]. See Figure 1. Recently, a $(2+\epsilon)$-factor approximation for split graphs was announced [23].

In this paper, we take a step towards the (affirmative) answer of the question of [2] by presenting a constant-factor approximation algorithm for the intersection of chordal and distance-hereditary graphs, known as *ptolemaic graphs*.[3] They are precisely graphs without any induced $C_{\geqslant 4}$ or a gem, so it is easy to see that they form a superclass of both 3-leaf power and block graphs.

Weighted Ptolemaic Deletion
Input : A graph $G = (V, E)$ with vertex weights $w : V \to \mathbb{R}^+ \cup \{0\}$.
Question : Find a set $S \subseteq V$ of minimum weight such that $G \backslash S$ is ptolemaic.

▶ **Theorem 1.1.** Weighted Ptolemaic Deletion *admits a polynomial-time constant-factor approximation algorithm.*

1.1 Techniques

Our proof presents new properties and algorithmic results on *inter-clique* digraphs as well as an approximation algorithm for a variant of Feedback Vertex Set that exploits this relationship (named Feedback Vertex Set with Precedence Constraints), each of which may be of independent interest.

[2] Any graph has either a vertex-disjoint packing of $k+1$ cycles, or a feedback vertex set of size $O(k \log k)$.
[3] The name *ptolemaic* comes from the fact that the shortest path distance satisfies *Ptolemy's inequality*: For every four vertices u, v, w, x, the inequality $d(u,v)d(w,x) + d(u,x)d(v,w) \geqslant d(u,w)d(v,x)$ holds.

1.1.1 Inter-clique Digraphs

The starting point of our proof is to examine what we call an *inter-clique digraph* of G. Let $\mathcal{C}(G)$ be the collection of all non-empty intersections of maximal cliques in G, see Section 2 for the formal definition. An inter-clique digraph $\overrightarrow{T}(G)$ of G, or simply $\overrightarrow{T}$, is a digraph isomorphic to the Hasse diagram of $(\mathcal{C}(G), \subseteq)$. A neat characterization of ptolemaic graphs was presented by Uehara and Uno [28]: a graph G is ptolemaic if and only if its inter-clique digraph is a forest. This immediately suggests the use of an $O(1)$-approximation algorithm for Feedback Vertex Set on the inter-clique digraph. Indeed, the black-box application of an $O(1)$-approximation algorithm for Feedback Vertex Set yields $O(1)$-approximation algorithms for subclasses of ptolemaic graphs including block graphs [1] and 3-leaf power graphs [5].

However, to leverage this characterization for Ptolemaic Deletion, two issues need to be addressed. First, a polynomial-time algorithm to construct an inter-clique digraph of the input graph G is needed, while the size of an inter-clique digraph can be exponentially large for general graphs. Second, even with the inter-clique digraph of polynomial size at hand, the application of Feedback Vertex Set remains nontrivial since (1) after deletion of vertices, the structure of the inter-clique digraph may drastically change, and (2) feedback vertex sets for the inter-clique digraph must satisfy additional constraints that a deletion of a node $C \in \mathcal{C}(G)$ must imply the deletion of all nodes reachable from it (because they are subsets of C in G). Addressing each of these issues boils down to understanding the properties of an inter-clique digraph and elaborating the relationship between the input graph and its inter-clique digraph.

For general graphs, their inter-clique digraphs are acyclic digraphs in which each node can be precisely represented by all sources that have a directed path to the node. It turns out that eliminating from G all induced subgraphs isomorphic to C_4 and gem is key to tackling the aforementioned issues. We show that any hole of G indicates the existence of a cycle in $\mathsf{Und}(\overrightarrow{T})$, and vice versa when G is (C_4, gem)-free (Lemmas 3.7-3.8). This in turn lets us to identify a variant of Weighted Feedback Vertex Set, termed Feedback Vertex Set with Precedence Constraints and defined in Section 1.1.2, which is essentially equivalent to Ptolemaic Deletion on G when it takes the inter-clique digraph of G as an input; see Proposition 3.11. Moreover, each subdigraph of $\overrightarrow{T}$ induced by the ancestors of any node v of $\overrightarrow{T}$ is a directed tree rooted at v, see Lemma 3.5. (Similar statement holds for the descendants of v.) This property is used importantly in analyzing our approximation for Feedback Vertex Set with Precedence Constraints. As Feedback Vertex Set with Precedence Constraints takes an inter-clique digraph as an input, we need to construct it in polynomial time. This is prohibitively time-consuming for general graphs. We show that the construction becomes efficient when G is both C_4 and gem-free, see Proposition 3.9.

1.1.2 Feedback Vertex Set with Precedence Constraints

Given acyclic directed graphs $\overrightarrow{G}$ and a vertex v, let $anc(v)$ and $des(v)$ be the set of ancestors and descendants respectively, and let $\mathsf{Und}(\overrightarrow{G})$ denote the underlying undirected graph of $\overrightarrow{G}$. It remains to design a constant-factor approximation algorithm for the following problem:

Feedback Vertex Set with Precedence Constraints (FVSP) **Input :** An acyclic directed graph $\overrightarrow{G} = (V, A)$, where each vertex v has weight $\omega_v \in \mathbb{R}^+ \cup \{0\}$. For each $v \in V$, the subgraph induced by $anc(v)$ is an in-tree rooted at v. **Question :** Delete a minimum-weight vertex set $S \subseteq V$ such that (1) $v \in S$ implies $des(v) \subseteq S$, (2) $\mathsf{Und}(\overrightarrow{G} \backslash S)$ is a forest.

It is a variant of Undirected Feedback Vertex Set (FVS) on $\mathsf{Und}(\overrightarrow{G})$, with the additional precedence constraint on S captured by directions of arcs in A. This precedence constraint makes an algorithm for FVSP harder to analyze than FVS because a vertex v can be deleted "indirectly"; even when v does not participate in any cycle, deletion of any ancestor of v forces to v to be deleted, so the analysis for v needs to keep track of every vertex in $anc(v)$.

We adapt a recent constant-factor approximation algorithm for Subset Feedback Vertex Set by Chekuri and Madan [10] for FVSP. The linear programming (LP) relaxation variables are $\{z_v\}_{v\in V}$, where z_v is supposed to indicate whether v is deleted or not, as well as $\{x_{ue}\}_{e\in A, u\in e}$, where x_{ue} is supposed to indicate that in the resulting forest $\mathsf{Und}(\overrightarrow{G}\backslash S)$ rooted at arbitrary vertices, whether e is the edge connecting u and its parent.

$$\text{Minimize } \sum_{v\in V} z_v \omega_v$$

$$\text{Subject to } z_v + x_{ue} + x_{ve} = 1 \text{ for each } e=(u,v)\in A, \quad z_v + \sum_{e\ni v} x_{ve} \leqslant 1 \text{ for each } v \in V,$$

$$z_u \leqslant z_v \text{ for each } e=(u,v)\in A, \qquad 0 \leqslant x, z \leqslant 1.$$

Compared to the LP in [10], we added the $z_u \leqslant z_v$ for all $(u,v)\in A$ to encode the fact that u's deletion implies v's deletion. This LP is not technically a relaxation, but one can easily observe that in any integral solution, the graph induced by $\{v : z_v = 0\}$ has at most one cycle, which can be easily handled later.[4] The rounding algorithm proceeds as follows. Fix three parameters $\varepsilon \approx 0.029, \alpha \approx 0.514, \beta \approx 0.588$. For notational convenience, let $\bar{x}_{ue} := 1 - x_{ue}$. Also, for each $e=(u,v)\in A$, let $y_e = z_v - z_u$.

(i) Delete all vertex v with $z_v \geqslant \varepsilon$.
(ii) Sample θ uniformly at random from the interval $[\alpha, \beta]$.
(iii) For each $e=(u,v)\in A$, if $\theta \in [\bar{x}_{ve} - y_e, \bar{x}_{ve}]$, delete $des(v)$.

Slightly modifying the analysis of [10], one can show that after rounding, there is indeed at most one cycle remained in each connected component. In terms of the total weight of deleted vertices, it is easy to bound the total weight of deleted vertices in Step (i) and the final cleanup step for one cycle. The main technical lemma of the analysis bounds the weight of vertices deleted in Step (iii) by at most $O(\mathsf{LP})$.

▶ **Lemma 1.2.** *For each $v \in V$, $\Pr[v$ is deleted in Step (iii)$] \leqslant O(z_v)$.*

Recall that $anc(v)$ induces the directed tree $\overrightarrow{T}$ rooted on v where all arcs are directed towards v, and deletion of any vertex in $\overrightarrow{T}$ forces the deletion of v. The lemma is proved by showing that while $anc(v)$ can be large, all vertices that can be possibly deleted during the rounding algorithm can be covered by at most two directed paths; it is proved by examining behaviors of the rounding algorithm on directed trees, followed by an application of Dilworth's theorem. The new LP constraint $z_u \leqslant z_v$ for all $(u,v)\in A$ ensures that the sum of the deletion probabilities along any path is at most $O(z_v)$, so the total probability that v is deleted can be bounded by $O(z_v)$.

[4] [10] added an additional cycle covering constraint in the LP. We find it conceptually easier to deal with the last remaining cycle separately at the end.

2 Preliminaries

For a mapping $f : X \to Y$ between two finite sets and a set $A \subseteq X$, we denote $\bigcup_{x\in A}\{f(x)\}$ by $f(A)$. For sets X and Y, we say that X and Y are *overlapping* if none of $X\backslash Y$, $Y\backslash X$, and $X \cap Y$ is empty. For a family $\mathcal{F}$ of sets, $\mathcal{F}$ is *laminar* if $\mathcal{F}$ has no overlapping two elements. In an undirected graph G, we say that two vertices u, v are *true twins*, or simply twins, if $N_G[u] = N_G[v]$. Note that true twins must be adjacent. Since the true twin relation is an equivalence relation, the true twin classes of V is uniquely defined. For graphs $G_1, \ldots, G_m$, we say that G is $(G_1, \ldots, G_m)$*-free* if G has no induced subgraph isomorphic to one of $G_1, \ldots, G_m$.

For a directed graph $\vec{G}$, we denote by $\mathsf{Und}(\vec{G})$ the underlying graph of $\vec{G}$. An *ancestor* of v in $\vec{G}$ is a vertex which is reachable to v in $\vec{G}$ and a *descendant* of v in $\vec{G}$ is a vertex which is reachable from v in $\vec{G}$. Two vertices u and v are *incomparable* in $\vec{G}$ if neither one is an ancestor of the other. A *source* of $\vec{G}$ is a vertex of $\vec{G}$ without an in-coming arc and a *sink* of $\vec{G}$ is a vertex without an out-going arc. We denote by $anc(\vec{G}, v)$ the set of ancestors of v in $\vec{G}$, by $des(\vec{G}, v)$ the set of descendants of v in $\vec{G}$, and by $src(\vec{G}, v)$ be the set of sources of $\vec{G}$ which are ancestors of v. When $\vec{G}$ is clear from the context, we may simply write $anc(v)$, $des(v)$, and $src(v)$, respectively. For an undirected cycle H in $\vec{H}$, we term a maximal directed subpath of G a *segment* of the cycle H. The *segment length* of a cycle H is defined as the number of segments of H. A *segment decomposition* of a cycle H is a cyclic sequence of all segments of H such that any two consecutive segments share a vertex of $\vec{G}$. We will write a segment decomposition of H as $H = x_0, \vec{P}_1, x_1, \overleftarrow{P}_2, x_2, \cdots, x_{2\ell-1}, \overleftarrow{P}_{2\ell}, x_{2\ell}(= x_0)$, in which for every odd i, $\vec{P}_i$ is a forward-oriented path from x_{i-1} to x_i and for every even i, P_i is a backward-oriented path from x_{i-1} to x_i.

2.1 Clique and inter-clique digraph

We denote the set of maximal cliques in a graph G by $\mathcal{M}(G)$. We define the set $\mathcal{C}(G)$ all non-empty intersections among maximal cliques, that is,

$$\mathcal{C}(G) := \bigcup_{\mathcal{I}\subseteq\mathcal{M}(G)} \left\{ C : C = \bigcap_{M\in\mathcal{I}} M,\ C \neq \varnothing \right\}.$$

We may write $\mathcal{M}(G)$ and $\mathcal{C}(G)$ as $\mathcal{M}$ and $\mathcal{C}$ respectively, if it is clear from the context.

Cleary, $\mathcal{C}(G)$ defines a partially ordered set under the set containment relation $\subseteq$. A *Hasse diagram* $\vec{H}$ of a poset $(S, \leqslant)$ represents each element of S as a vertex and adds an arc from y to x if and only if $y > x$ and there is no element $z \in S$ with $y > z > x$. We say that a digraph $\vec{T}$ is an *inter-clique digraph* of G if $\vec{T}$ isomorphic to the Hasse diagram of the poset $(\mathcal{C}(G), \subseteq)$. For an inter-clique digraph $\vec{T}$ of G or the Hasse diagram $\vec{H}$, we call $V(\vec{T})$ or $V(\vec{H})$ *nodes* instead of vertices in order to distinguish them from the vertices of G.

For a vertex set $X \subseteq V(G)$, we define $src(X)$ as the set of all maximal cliques containing X. For $v \in V(G)$, we may write $src(v)$ instead of $src(\{v\})$. For a collection of sets $\mathcal{X}$, $src(\mathcal{X})$ is defined as the collection of sets (without duplicates) $src(\mathcal{X}) = \{src(X) : X \in \mathcal{X}\}$. For the Hasse diagram $\vec{H}$ of $(\mathcal{C}(G), \subseteq)$ and a clique $C \in \mathcal{C}(G)$, we have $src(C) = src(\vec{H}, C)$, and this justifies the reuse of the notation src for a vertex set, while $src(\vec{G}, v)$ is already defined to delineate the set of vertices with no in-coming arcs from which there is a directed path to v in $\vec{G}$.

2.2 Ptolemaic graphs

A graph is *ptolemaic* if for every four vertices a, b, c, and d in the same connected component, G satisfies the following inequality:

$$\mathrm{dist}_G(a,b) \cdot \mathrm{dist}_G(c,d) \leqslant \mathrm{dist}_G(a,c) \cdot \mathrm{dist}_G(b,d) + \mathrm{dist}_G(a,d) \cdot \mathrm{dist}_G(b,c).$$

Note that they can be equivalently defined as the set of ($C_{\geqslant 4}$, gem)-free graphs. The following theorem proves strong relationship between ptolemaic graphs and its inter-clique digraph.

▶ **Theorem 2.1** (Uehara and Uno [28]). *A graph G is ptolemaic if and only if* $\mathsf{Und}(\overrightarrow{H})$ *is a forest, where $\overrightarrow{H}$ is the Hasse diagram of $(\mathcal{C}(G), \subseteq)$.*

3 Structures of Inter-clique digraphs

3.1 Basic properties of inter-clique digraphs

In this subsection, we investigate the properties of the Hasse diagram $\overrightarrow{H}$ of the poset $(\mathcal{C}(G), \subseteq)$ for a graph $G = (V, E)$. All the results presented in this subsection assume no restriction on the input graph G.

We first observe that each vertex v of V can be uniquely associated to a clique C of $\mathcal{C}(G)$ with the property $src(v) = src(C)$.

▶ **Lemma 3.1.** *For every vertex v of G, there is a unique minimal element $C(v) \in \mathcal{C}(G)$ containing v in the poset $(\mathcal{C}(G), \subseteq)$ and it holds that $C(v) = \bigcap_{M \in src(v)} M$.*

We call the clique as depicted in Lemma 3.1 the *canonical clique* of v, namely the canonical clique is defined as $C(v) = \bigcap_{M \in src(v)} M$. Note that $src(v) = src(C(v))$.

In the following three lemmas, we investigate properties of descendants of nodes in $\overrightarrow{H}$. In this extended abstract, the proofs of some lemmas will be deferred to the full version.

▶ **Lemma 3.2.** *If a node C has immediate descendants $C_1, \ldots, C_p$ with $p \geqslant 2$ in $\overrightarrow{H}$, then we have $src(C) = src(C_i) \cap src(C_j)$ for every $1 \leqslant i < j \leqslant p$.*

▶ **Lemma 3.3.** *Let Z be a true twin class of G contained in a clique $C \in \mathcal{C}(G)$. Then the following are equivalent.*

(i) *$src(C) \subsetneq src(Z)$.*

(ii) *There exists a proper descendant C' of C in $\overrightarrow{H}$ such that $Z \subseteq C'$.*

▶ **Lemma 3.4.** *Let C_1 and C_2 be two cliques of $\mathcal{C}(G)$. Then $\overrightarrow{H}$ contains at most one greatest common descendant of C_1 and C_2.*

3.2 Inter-clique digraphs of (C_4, gem)-free graphs

Here, we examine how the extra assumption that G is (C_4, gem)-free brings about a new structure to emerge in the corresponding Hasse diagram $\overrightarrow{H}$.

▶ **Lemma 3.5.** *Let $G = (V, E)$ be a (C_4, gem)-free graph and M be a maximal clique of G. Then $\mathcal{C}_M := \{C \in \mathcal{C}(G) : C \subseteq M\}$ is laminar and $\overrightarrow{H}[\mathcal{C}_M]$ is an out-tree rooted at M.*

The following two lemmas will be crucially used in the proof of Proposition 3.11 to investigate the structure of minimal ptolemaic deletion set.

▶ **Lemma 3.6.** *Let $G = (V, E)$ be a (C_4, gem)-free graph. If G has a hole H and $v \in V(H)$, then $G[V(H) \cup \{v'\} \backslash \{v\}]$ contains a hole for every $v' \in C(v)$.*

▶ **Lemma 3.7.** *Let $G = (V, E)$ be a (C_4, gem)-free graph. Then any undirected cycle H of $\overrightarrow{H}$ has segment length at least 8.*

Sketch of the proof. Let $H = C_0, \vec{P}_1, C_1, \overleftarrow{P}_2, C_2, \cdots, C_{2\ell-1}, \overleftarrow{P}_{2\ell}, C_{2\ell}(= C_0)$ be a segment decomposition of H. We skip to prove the statement for $\ell \leqslant 2$. Suppose that $\ell = 3$, and note that C_{2i} is a common ancestor of C_{2i-1} and C_{2i+1} for every $i \in [3]$. For each $i \in [3]$, choose an arbitrary clique C'_i which is a sink in $\overrightarrow{H}$ and a descendant of C_{2i-1}. Then it is easy to see that C'_i is a descendant of C_{2i-1} only for each i. On the other hand, the cliques C_{2i-1} and C_{2i+1} are completely adjacent for every $i \in [3]$, which implies that $C'_1 \cup C'_2 \cup C'_3$ is a clique because $C'_i \subseteq C_{2i-1}$ for each $i \in [3]$. Consider a maximal clique M containing $C'_1 \cup C'_2 \cup C'_3$ and note that all the nodes of H are descendants of M in $\overrightarrow{H}$. This contradicts Lemma 3.5, which asserts that $\overrightarrow{H}[\mathcal{C}_M]$ is an out-tree rooted at M, where $\mathcal{C}_M := \{C \in \mathcal{C}(G) : C \subseteq M\}$. This completes the proof of claim. ◀

From the laminar structure of $\overrightarrow{H}[\mathcal{C}_M]$, we can observe that any pair of nodes are incomparable in $\overrightarrow{H}$ if they do not belong to the same segment.

▶ **Lemma 3.8.** *Let H be a cycle of $\overrightarrow{H}$ with the shortest segment length with a segment decomposition $H = C_0, \vec{P}_1, C_1, \overleftarrow{P}_2, C_2, \cdots, C_{2\ell-1}, \overleftarrow{P}_{2\ell}, C_{2\ell}(= C_0)$. Then for any two nodes C, C' of H, C and C' are incomparable unless they belong to the same segment of H. In addition, for $i, j \in [\ell]$ with $|i - j| \geqslant 2$, there is no common ancestor of C_{2i-1} and C_{2j-1} in $\overrightarrow{H}$.*

3.3 Constructing inter-clique digraphs for (C_4, gem)-free graphs

As an arbitrary graph can have prohibitively many maximal cliques, we cannot expect a polynomial-time algorithm to construct inter-clique digraphs for general graphs. Instead, we present a polynomial-time algorithm for (C_4, gem)-free graphs. Let $\mathcal{C}_M := \{C \in \mathcal{C} : M \in \mathit{src}(C)\}$ for each $M \in \mathcal{M}$.

▶ **Proposition 3.9.** *There is a polynomial-time algorithm which, given a (C_4, gem)-free graph G, constructs the Hasse diagram $\overrightarrow{H}$ of $(\mathcal{C}(G), \subseteq)$.*

Sketch of the proof. Let $\mathcal{Z}$ be the partition of V into true twin classes and $n := |V(G)|$. Since G has no C_4 as an induced subgraph, it has at most n^2 maximal cliques and these cliques can be enumerated with polynomial delay [12, 27]. Thus, all of $\mathcal{M}$, $\mathcal{Z}$, and $\mathit{src}(\mathcal{Z})$ can be computed in polynomial time. Observe that for certain cliques $C \in \mathcal{C}(G)$, $\mathit{src}(C)$ is already contained in $\mathit{src}(\mathcal{Z})$. By Lemma 3.3, we can easily show that if $C \in \mathcal{C}(G)$ is a sink or has a unique immediate descendant in $\overrightarrow{H}$, then $\mathit{src}(C) \in \mathit{src}(\mathcal{Z})$. Let $\mathcal{R}^{\cap,0} := \mathit{src}(\mathcal{Z})$. For $i \geqslant 1$, we define $\mathcal{R}^{\cap,i}$ recursively as $\mathcal{R}^{\cap,i} := \mathcal{R}^{\cap,i-1} \cup \{R \cap R' : R, R' \in \mathcal{R}^{\cap,i-1}\}$. Let the *height of a node* v of an acyclic digraph $\overrightarrow{G}$ be the length of a longest directed path from v to a sink in $\overrightarrow{G}$. The height of $\overrightarrow{G}$ is defined as the maximum over the heights of all nodes of $\overrightarrow{G}$.

▷ Claim 3.10. $\mathcal{R}^{\cap,s} = \mathit{src}(\mathcal{C}(G))$, where s is the height of $\overrightarrow{H}$.

Proof. It suffices to prove the following for each $i \geqslant 0$: for any node C at height i in $\overrightarrow{H}$, we have $\mathit{src}(C) \in \mathcal{R}^{\cap,i}$. By the previous paragraph, we only need to consider a node C such that C has height $i > 0$ and (at least) two immediate descendants C_1, C_2 in $\overrightarrow{H}$. By induction hypothesis and because of that the height of the immediate descendants of C is at most $i - 1$, we have $\mathit{src}(C_1), \mathit{src}(C_2) \in \mathcal{R}^{\cap,i-1}$. Therefore, we have $\mathit{src}(C_1) \cap \mathit{src}(C_2) \in \mathcal{R}^{\cap,i}$ by definition. Then by Lemma 3.2, it holds that $\mathit{src}(C) \in \mathcal{R}^{\cap,i}$ as claimed. ◁

By Lemma 3.5, we can show that for each maximal clique M, the height of $\overrightarrow{H}[\mathcal{C}_M]$ is at most $|\mathcal{Z}|$, and therefore the height of $\overrightarrow{H}$ is at most $|\mathcal{Z}|$, that is, at most n. As we compute $\mathcal{R}^{\cap,i+1}$ from $\mathcal{R}^{\cap,i}$ repeatedly, we need a guarantee that the sizes of the computed sets $\mathcal{R}^{\cap,i}$

do not grow exponentially. For each maximal clique M, by the laminarity of $\mathcal{C}_M$, we can show that $|\mathcal{C}_M| \leqslant 2n$, and therefore $|\mathcal{C}(G)| \leqslant 2n^3$. Then we can compute each $\mathcal{R}^{\cap,i}$ in polynomial time and $src(\mathcal{C}(G))$ can be computed in polynomial time by Claim 3.10. As we compute $\mathcal{R}^{\cap,i}$, the containment relations amongst the elements of $\mathcal{R}^{\cap,i}$ can be determined as well. Then $\overrightarrow{H}$ obviously comes from the Hasse diagram of $(src(\mathcal{C}(G)), \subseteq)$. ◀

3.4 Reduction from Ptolemaic Deletion to Feedback Vertex Set with Precedence Constraints

Let $G = (V, E)$ be a (C_4, gem)-free graph with weight $\omega^o : V \to \mathbb{R}_+ \cup \{0\}$, $\overrightarrow{H}$ be the Hasse diagram of $(\mathcal{C}(G), \subseteq)$, and $\overrightarrow{T} = (N, A)$ be an inter-clique digraph isomorphic to $\overrightarrow{H}$ with an arc-preserving mapping $\gamma : \mathcal{C}(G) \to N$. Notice that the canonical clique can be construed as a function which maps each vertex v of G to the clique $C \in \mathcal{C}(G)$ such that $src(v) = src(C)$. We define a mapping $C^{-1} : \mathcal{C}(G) \to 2^V$ so that it maps each clique C of $\mathcal{C}(G)$ to its preimage under the canonical clique as a function from V to $\mathcal{C}(G)$: if there is no vertex $v \in V$ with $C(v) = C$, then the preimage of C under the canonical clique is $\varnothing$. We define $\phi : V \to N$ and $\phi^{-1} : N \to 2^V$ such that $\phi(v) = \gamma(C(v))$ and $\phi^{-1}(x) = C^{-1}(\gamma^{-1}(x))$. Now the node weight function $\omega : N \to \mathbb{R}_+ \cup \{0\}$ is defined as $\omega(x) := \sum_{v \in \phi^{-1}(x)} \omega^o(v)$.

For a set of nodes R of $\overrightarrow{T}$, the *closure* of R, denoted as R^*, is a minimal superset of R for which the following holds:

(a) all descendants of R of weight zero are contained in R^*,
(b) if all immediate descendants of a node v are in R^* and $\phi^{-1}(v) = \varnothing$, then $v \in R^*$.

▶ **Proposition 3.11.** *Let $G = (V, E)$ be a (C_4, gem)-free graph with vertex weight $\omega^o : V \to \mathbb{R}$. Let $\overrightarrow{T} = (N, A)$ be an inter-clique digraph of G with an arc-preserving mapping $\gamma : \mathcal{C}(G) \to N$ and with node weight $\omega : \mathcal{C} \to \mathbb{R}_+ \cup \{0\}$, such that $\omega(x) := \sum_{v \in \phi^{-1}(x)} \omega^o(v)$. Then the following two statements hold.*

1. *For any minimal ptolemaic deletion set $S \subseteq V$, (i) $\phi(S)^*$ is downward-closed in $\overrightarrow{T}$, (ii) $\mathsf{Und}(\overrightarrow{T} \backslash \phi(S)^*)$ is a forest, and (iii) $\sum_{x \in \phi(S)^*} \omega(x) = \sum_{v \in S} \omega^o(v)$*
2. *For any $R \subseteq N$ such that (i) R is downward-closed in $\overrightarrow{T}$, and (ii) $\mathsf{Und}(\overrightarrow{T} \backslash R)$ is a forest, $\phi^{-1}(R)$ is a ptolemaic deletion set of G of weight $\sum_{x \in R} \omega(x)$.*

Sketch of the Proof. We first prove (1)-(i). We first observe that if S is a minimal deletion set, S contains the canonical clique $C(v)$ of v whenever S contains $v \in V$.

▷ Claim 3.12. If $S \subseteq V$ is a minimal ptolematic deletion set, then $C(v) \subseteq S$ whenever $v \in S$. Consequently, $\phi^{-1}(x) \subseteq S$ for every $x \in \phi(S)$.

Proof. Suppose $C(v) \nsubseteq S$ for some $v \in S$. Since G is (C_4, gem)-free, $G \backslash S$ is ptolemaic if and only if $G \backslash S$ is chordal. Since S is minimal, $G \backslash (S \backslash \{v\})$ has a hole H intersecting v. By the assumption, there exists $v' \in C(v) \backslash S$. However, Lemma 3.6 implies that $G[(V(H) \backslash \{v\}) \cup \{v'\}]$ contains a hole and thus $G \backslash S$ contains a hole, a contradiction. The second statement is immediate from the first statement. ◁

Consider a vertex $v \in S$ of G and an arbitrary descendant x of $\phi(v)$ in $\overrightarrow{T}$. We claim that $x \in \phi(S)^*$. If $\phi^{-1}(x) = \varnothing$, then by definition $\omega(x) = \sum_{v \in \varnothing} \omega^o(v) = 0$ and thus the claim trivially holds by definition of $\phi(S)^*$. Otherwise, let $w \in \phi^{-1}(x)$ and we have $\phi^{-1}(x) \subseteq \gamma^{-1}(x) \subseteq \gamma^{-1}(\phi(v)) = C(v) \subseteq S$. Thus, $w \in \phi^{-1}(x) \subseteq S$ which implies $x \in \phi(S)$, and $\phi(S)^*$ is downward-closed in $\overrightarrow{T}$.

To see that (1)-(ii), let H be a cycle of $\overrightarrow{T}\backslash\phi(S)^*$ with the least segment length and let $x_0, \vec{P}_1, x_1, \overleftarrow{P}_2, x_2, \cdots, x_{2\ell-1}, \overleftarrow{P}_{2\ell}, x_{2\ell}(= x_0)$ be a segment decomposition of H. Consider the cliques $\gamma^{-1}(x_{2i-1})\backslash S$ of G for $i \in [\ell]$. We can show by (1)-(i) and the definition of the closure of a node set, that for every $i \in [\ell]$, there exists a vertex $v_i \in \gamma^{-1}(x_{2i-1})\backslash S$ of G.

We observe that all v_i's are distinct. Suppose that $v_i = v_j$ for $i \neq j$, and without loss of generality we may assume that $1 \leqslant i < j \leqslant \ell$. Then the canonical clique $C(v_i)$ is a common descendant of $\gamma^{-1}(x_{2i-1})$ and $\gamma^{-1}(x_{2j-1})$, or equivalently, $\phi(v_i)$ is a common descendant of x_{2i-1} and x_{2j-1}. Let x^* be the greatest common descendant of x_{2i-1} and x_{2j-1} in $\overrightarrow{T}$, which is unique by Lemma 3.4. Let P and Q be the directed (x_{2i-1}, x^*)-path and the directed (x_{2j-1}, x^*)-path. Due to Lemma 3.8, both directed paths are disjoint from H except from the two starting vertex x_{2i-1} and x_{2j-1}. Then we can find a cycle from H with a shorter segment length by replacing a subpath of H between x_{2i-1} and x_{2j-1} with P, x^*, Q, a contradiction.

Furthermore, v_i and v_{i+1} are adjacent because the cliques $\gamma^{-1}(x_{2i-1})$ and $\gamma^{-1}(x_{2(i+1)-1})$ are complete to each other in G due to the existence of common ancestor x_{2i} in $\overrightarrow{T}$. That is, $J = v_1, \ldots, v_\ell, v_1$ forms a cycle, and its length is at least four by Lemma 3.7. Furthermore, Lemma 3.8 implies that J is a hole, which altogether avoids S because of our choice of v_i as a vertex of $\gamma^{-1}(x_{2i-1})\backslash S$. This contradicts the assumption that S is a ptolemaic deletion set, which proves (1)-(ii). We skip to prove (1)-(iii).

To see (2), suppose that for a node set R of $\overrightarrow{T}$ (i) R is downward-closed in $\overrightarrow{T}$, and (ii) $\mathsf{Und}(\overrightarrow{T}\backslash R)$ is a forest while $\bigcup_{x\in R}\phi^{-1}(x)$ is not a ptolemaic deletion set of G. Let $H = v_1, \ldots, v_s, v_1$ be a hole of length $s \geqslant 5$ in $G\backslash\bigcup_{x\in R}\phi^{-1}(x)$. Consider the canonical cliques $C(v_1), \ldots, C(v_s)$ and their corresponding nodes $x_1, \ldots, x_s$ in $\overrightarrow{T}$. The adjacency of v_i and v_{i+1} ensures that x_i and x_{i+1} has a common ancestor for all $i \in [s]$, where $s + 1 = 1$. Furthermore, none of the nodes from these common ancestors is contained in R since otherwise, some x_i must belong to the downward-closed set R. This, however, means that $x_1, \ldots, x_s$ are contained in a closed walk of $\overrightarrow{T}\backslash R$, contradicting (ii). We conclude that $\bigcup_{x\in R}\phi^{-1}(x)$ is a ptolemaic deletion set of G and its weight is easily computed as suggested. ◀

▶ **Theorem 3.13.** *There is a polynomial-time algorithm which, given a graph $G = (V, E)$ with vertex-weight $\omega^o : V \to \mathbb{R}_+ \cup \{0\}$, returns a ptolemaic deletion set $S \subseteq V$ of weight at most $68 \cdot \mathsf{OPT}_{pto}$, where OPT_{pto} is the minimum weight of a ptolematic deletion set of G.*

Sketch of the proof. We skip the trivial runtime analysis. In order to turn the input graph into a (C_4, gem)-free graph, we can design a simple linear programming (LP) to hit all C_4 and gem, and let X be the set of vertices whose LP value is at least 1/5. Since every copy of C_4 (resp. gem) must have a vertex with LP value at least 1/4 (resp. 1/5), $G' := G\backslash X$ is (C_4, gem)-free. Furthermore, the total weight of X is at most 5 times the LP value, which is at most $5\mathsf{OPT}_{pto}$.

Each vertex of G' inherits its weight ω^o_v in G. We construct an inter-clique digraph $\overrightarrow{T} = (N, A)$ of G' with a node-weight ω as in Proposition 3.11. The node set $anc(x)$ forms an in-tree rooted at x due to Lemma 3.5, which means that $(\overrightarrow{T}, \omega)$ is a legitimate instance to Feedback Vertex Set with Precedence Constraints. Then by Theorem 4.1, we get a solution $R \in N$ such that R is downward-closed in $\overrightarrow{T}$, $\mathsf{Und}(\overrightarrow{T}\backslash R)$ is a forest, and $\omega(R) \leqslant 63\mathsf{OPT}_{fvsp}$, where OPT_{fvsp} is the minimum weight of a solution to Feedback Vertex Set with Precedence Constraints. Since $\mathsf{OPT}_{fvsp} \leqslant \mathsf{OPT}_{pto}$ by (1) of Proposiotion 3.11, $\bigcup_{x\in R}\phi^{-1}(x) \cup X$ is a desired ptolemaic deletion set of G. ◀

4 Constant-factor approximation algorithm

In this section, we consider Feedback Vertex Set with Precedence Constraints introduced in Section 1.1.2.

> FEEDBACK VERTEX SET WITH PRECEDENCE CONSTRAINTS
> **Input :** An acyclic directed graph $\overrightarrow{G} = (V, A)$, where each vertex v has weight $\omega_v \in \mathbb{R}^+ \cup \{0\}$. For each $v \in V$, the subgraph induced by $anc(v)$ is an in-tree rooted at v.
> **Question :** Delete a minimum-weight vertex set $S \subseteq V$ such that (1) $v \in S$ implies $des(v) \subseteq S$, (2) $\mathsf{Und}(\overrightarrow{G} \backslash S)$ is a forest.

It is a variant of UNDIRECTED FEEDBACK VERTEX SET on $\mathsf{Und}(\overrightarrow{G})$, with the additional precedence constraint on S is captured by the direction of arcs in A. The main result of this section is an $O(1)$-approximation algorithm for this problem.

▶ **Theorem 4.1.** *There is a polynomial-time* 63*-approximation algorithm for* FEEDBACK VERTEX SET WITH PRECEDENCE CONSTRAINTS.

We consider the following linear programming (LP) relaxation. The relaxation variables are $\{z_v\}_{v \in V}$, where z_v is supposed to indicate whether v is deleted or not, as well as $\{x_{ue}\}_{e \in A, u \in e}$, where x_{ue} is supposed to indicate that in the resulting forest $\mathsf{Und}(\overrightarrow{G} \backslash S)$ rooted at arbitrary vertices, whether e is the edge connecting u and its parent.

$$\text{Minimize} \sum_{v \in V} \omega_v z_v$$

$$\text{Subject to } z_v + x_{ue} + x_{ve} = 1 \qquad \forall e = (u, v) \in A \qquad (1)$$

$$z_v + \sum_{e \ni v} x_{ve} \leqslant 1 \qquad \forall v \in V \qquad (2)$$

$$z_u \leqslant z_v \qquad \forall e = (u, v) \in A$$

$$0 \leqslant x, z \leqslant 1.$$

Let OPT be the weight of the optimal solution, and $\mathsf{LP} \leqslant \mathsf{OPT}$ be the optimal value of the above LP. After solving the LP, we perform the following rounding algorithm. It is parameterized by three parameters $\varepsilon, \alpha, \beta \in (0, 1)$ that satisfy

$$2\alpha \geqslant 1 + \varepsilon, \qquad (3)$$

$$3(1 - \beta) \geqslant 1 + 8\varepsilon. \qquad (4)$$

(The final choice will be $\varepsilon \approx 0.029, \alpha \approx 0.514, \beta \approx 0.588$.) For notational convenience, let $\bar{x}_{ue} := 1 - x_{ue}$. Also, for each $e = (u, v) \in A$, let $y_e = z_v - z_u$. Each vertex $v \in V$ maintains a set $L_v \subseteq A$. Initially, all L_v's are empty.

(i) Delete all vertex v with $z_v \geqslant \varepsilon$.
(ii) Sample θ uniformly at random from the interval $[\alpha, \beta]$.
(iii) For each $e = (u, v) \in A$,
- If $\theta \in [\bar{x}_{ve} - y_e, \bar{x}_{ve}]$, delete $des(v)$. Say v is *directly deleted by* e.
- Otherwise,
  - If $\theta > \bar{x}_{ve}$, then add e to L_v and say v *points to* e.
  - If $\theta > \bar{x}_{ue}$, then add e to L_u and say u *points to* e.

Though the above rounding algorithm is stated as a randomized algorithm, it is easy to make it deterministic, because there are at most $O(m)$ subintervals of $[\alpha, \beta]$ such that two θ values from the same interval behave exactly the same in the rounding algorithm.

We first analyze the total weight of deleted vertices. In Step (i), we delete all vertices whose LP value $z_v \geqslant \varepsilon$, so the total weight of deleted vertices in Step (i) is at most LP/ε. The following lemma bounds the weight of vertices deleted in Step (iii) by at most $2\mathsf{LP}/(\beta - \alpha)$.

▶ **Lemma 4.2.** *For each* $v \in V$*,* $\Pr[v$ *is deleted in Step (iii)*$] \leqslant \frac{2z_v}{\beta-\alpha}$.

Proof. Due to Step (i), we can assume that every vertex v satisfies $z_v < \varepsilon$ and each arc e satisfies $y_e < \varepsilon$.

Fix a vertex $v \in V$. Let $\overrightarrow{T} = (V(\overrightarrow{T}), A(\overrightarrow{T}))$ be the subgraph of $\overrightarrow{G}$ induced $anc(v)$. By the definition of Feedback Vertex Set with Precedence Constraints, $\overrightarrow{T}$ is an in-tree rooted at v. We first prove the following claim that if we consider any directed path $(u_0, \dots, u_k)$ of $\overrightarrow{T}$ and the value of $x_{u_i,(u_{i-1},u_i)}$ that u_i gives to its incoming edge (u_{i-1}, u_i), the value at the end ($i = k$) is almost as large as the value at the beginning ($i = 1$).

▷ Claim 4.3. Let $(u_0, \dots, u_k)$ be a directed path in $\overrightarrow{T}$ and $e_i = (u_{i-1}, u_i)$. Then for any $i \in [k]$, $x_{u_ie_i} \geqslant x_{u_1e_1} - (z_{u_i} - z_{u_1}) \geqslant x_{u_1e_1} - \varepsilon$.

Proof. The proof proceeds by induction. The base case $i = 1$ is obviously true. When the claim holds for $i-1$, the constraint (2) of the LP (for u_{i-1}) implies

$$x_{u_{i-1}e_{i-1}} + z_{u_{i-1}} + x_{u_{i-1}e_i} \leqslant 1,$$

and the constraint (1) of the LP implies (for e_i)

$$z_{u_i} + x_{u_{i-1}e_i} + x_{u_ie_i} = 1.$$

Subtracting the first inequality from the second equality yields

$$x_{u_ie_i} \geqslant x_{u_{i-1}e_{i-1}} - (z_{u_i} - z_{u_{i-1}}),$$

which, by the induction hypothesis, is at least

$$x_{u_1e_1} - (z_{u_{i-1}} - z_{u_1}) - (z_{u_i} - z_{u_{i-1}}) = x_{u_1e_1} - (z_{u_i} - z_{u_1}).$$ ◁

For $e = (w, u) \in A(\overrightarrow{T})$, call e a *target* if $\Pr[u$ is directly deleted by $e] > 0$, which implies $\bar{x}_{ue} - y_e < \beta \Rightarrow x_{ue} > 1 - \beta - y_e > 1 - \beta - \varepsilon$. For two arcs $e, f \in A(\overrightarrow{T})$, say they are *incomparable* if there is no directed path from the tail of one arc to tail of the other in $\overrightarrow{T}$ (though they may share the head.)

▷ Claim 4.4. There are no three pairwise incomparable targets.

Proof. Assume towards contradiction that there exist three pairwise incomparable targets $e_1 = (w_1, u_1), e_2 = (w_2, u_2), e_3 = (w_3, u_3)$. It implies that $x_{u_ie_i} > 1 - \beta - \varepsilon$ for each i. By Claim 4.3, for any i and any arc $e' = (w', u') \in A(\overrightarrow{T})$ that has a directed path from e_i, we have

$$x_{u'e'} > x_{u_ie_i} - \varepsilon > 1 - \beta - 2\varepsilon. \tag{5}$$

For each $i \in [3]$, consider the path P_i from w_i to v, and let g_i be the last arc of P_i that does not appear in any other P_j's. We consider the following two cases depending on how they intersect, and show both cannot happen.

First, suppose all g_1, g_2, g_3 meet at the same vertex w; in other words, $g_i = (t_i, w)$ for some t_i's. Then by (5), $x_{wg_i} > 1-\beta-2\varepsilon$ for each i. With (4), it implies $\sum_i x_{wg_i} > 3(1-\beta)-6\varepsilon \geqslant 1$, which violates the constraint (2) of the LP.

Finally, without loss of generality, suppose g_1 and g_2 meet at u, which is not incident on g_3; in other words, $g_1 = (t_1, u), g_2 = (t_2, u), g_3 = (t_3, w)$ for some t_i's, where w is an ancestor of u in $\overrightarrow{T}$ and is the first vertex where all P_1, P_2, P_3 intersect. Let $\overrightarrow{T}$ be the parent of u in the tree $\overrightarrow{T}$ ($\overrightarrow{T}$ may be equal to w), and $g = (u, t)$. Then (5), implies $x_{ug_i} > 1 - \beta - 2\varepsilon$ for $i \in \{1, 2\}$, which, combined with the LP constraint (2) for u, yields

$$x_{ug} < 1 - 2(1 - \beta - 2\varepsilon) = 2\beta - 1 + 4\varepsilon.$$

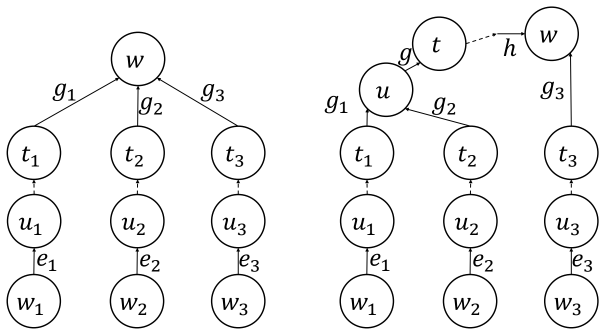

Figure 2 Two cases for g_1, g_2, g_3. The left figure shows the case when they all meet at the same vertex w. The right figure shows when g_1 and g_2 meet first at u and meet g_3 with w later. Real lines indicate an individual arc and dotted lines indicate a directed path.

Together again with the LP constraint (1) for g, we have

$$x_{tg} > 1 - x_{ug} - z_t > 2 - 2\beta - 5\varepsilon.$$

Let h be the last arc of the path from u to w. Using Claim 4.3 again, we conclude that $x_{wh} > 2 - 2\beta - 6\varepsilon$. Combined with $x_{wg_3} > 1 - \beta - 2\varepsilon$ and h and g_3 are different, it implies $x_{wh} + x_{wg_3} > 3 - 3\beta - 8\varepsilon \geqslant 1$ by (4), which contradicts the constraint (2) of the LP for w. ◁

Now we compute the probability that v is deleted by Step (iii) of the rounding algorithm. It happens whether v itself is directly deleted or some vertex $u \in anc(v) = V(\overrightarrow{T})$ is directly deleted by a target $e = (w, u)$. By Claim 4.4, no three targets are pairwise comparable, and by Dilworth's Theorem, all targets are contained in two directed paths P_1, P_2 in $\overrightarrow{T}$. By the choice of the rounding algorithm, for one path $P_1 = (u_0, \dots, u_k = v)$, for each $i \in [k]$,

$$\Pr[u_i \text{ is directly deleted by } (u_{i-1}, u_i)] \leqslant \frac{y_{(u_{i-1},u_i)}}{\beta - \alpha} = \frac{z_{u_i} - z_{u_{i-1}}}{\beta - \alpha}.$$

Summing over all i's yields

$$\sum_{i=1}^{k} \frac{z_{u_i} - z_{u_{i-1}}}{\beta - \alpha} = \frac{z_{u_k} - z_{u_0}}{\beta - \alpha} \leqslant \frac{z_v}{\beta - \alpha}.$$

We can apply the same analysis to P_2 and use the union bound. ◀

We now examine structure of the remaining graph after the rounding procedure. We first show that in the original graph, each arc, if not deleted, is pointed to by at least one of its endpoints.

▷ Claim 4.5. For each $e = (u, v) \in A$, if neither u nor v was deleted during the rounding, e is pointed to by at least one of them.

Proof. Since v is not deleted, it means $z_v < \varepsilon$, which, by (1), implies that $x_{ue} + x_{ve} > 1 - \varepsilon \Leftrightarrow \bar{x}_{ue} + \bar{x}_{ve} < 1 + \varepsilon$. Since $\theta \geqslant \alpha$, by (3), either $\theta \geqslant \bar{x}_{ue}$ or $\theta \geqslant \bar{x}_{ue}$. ◁

The following lemma shows that after the rounding, each connected component (in the undirected sense) has at most one cycle.

▶ **Lemma 4.6.** *Let S be the set of vertices deleted during the rounding algorithm. In each connected component of* $\mathsf{Und}(\overrightarrow{G}\backslash S)$*, there is at most one (undirected) cycle.*

Proof. The proof proceeds by examining how vertices can possibly point to adjacent arcs. First, the following claim shows that one vertex cannot point to more than two arcs.

▷ Claim 4.7. Every vertex $v \in V$ points to at most two arcs.

Proof. Assume towards contradiction that v points to three arcs e, f, and g. It implies $\bar{x}_{ve}, \bar{x}_{vf}, \bar{x}_{vg}$ are all strictly less than $\theta \leqslant \beta$, which implies that $x_{ve} + x_{vf} + x_{vg} > 3(1-\beta)$. Since $3(1-\beta) \geqslant 1$ by (4), it contradicts the constraint (2) of the LP relaxation. ◁

Moreover, the following claim constrains the way arcs in a cycle are pointed to by its vertices.

▷ Claim 4.8. For every arc $e \in A$, if it is pointed to by exactly one of its endpoint, say v, then it is the only arc that v points to.

Proof. We first show $\theta < x_{ve} + z_v$. If $e = (u, v)$, the assumption that u does not point to e implies

$$\theta < \bar{x}_{ue} = 1 - x_{ue} = 1 - (1 - x_{ve} - z_v) = x_{ve} + z_v,$$

where the second equality follows from (1). Even when $e = (v, u)$, the assumption that e is not deleted and u does not point to e implies

$$\theta < \bar{x}_{ue} - y_e = \bar{x}_{ue} - (z_u - z_v) = 1 - x_{ue} - z_u + z_v = x_{ve} + z_v,$$

where the last equality follows from the constraint (1) of the LP relaxation. Therefore, $\theta < x_{ve} + z_v$ in any case.

If v points to any other arc f, it implies

$$\theta > \bar{x}_{vf} = 1 - x_{vf} \geqslant z_v + x_{ve},$$

where the inequality follows from the constraint (2) of the LP relaxation. This leads to contradiction, proving the claim. ◁

Therefore, after the rounding, in the remaining graph $\overrightarrow{G}\backslash S$, (i) every remaining arc is pointed by at least one of its endpoints, (ii) each vertex points to at most two arcs, and (iii) if one vertex does not point to an arc incident on it, the other endpoint uniquely points to the arc.

Consider an undirected cycle $(v_1, \ldots, v_k, v_{k+1})$ in $\mathsf{Und}(\overrightarrow{G}\backslash S)$ with $v_1 = v_{k+1}$, so that either (v_i, v_{i+1}) or (v_{i+1}, v_i) is in A for every $i \in [k]$. Let $\{v_i, v_{i+1}\}$ denotes an undirected edge. If an edge in this cycle is pointed to by only one of its endpoints (without loss of generality, say $\{v_k, v_1\}$ is only pointed to by v_1), then v_1 cannot point to any other edge, so $\{v_2, v_1\}$ is uniquely pointed to by v_2 by (iii), and this inductively leads to every $\{v_i, v_{i+1}\}$ uniquely pointed to by v_{i+1} for $1 \leqslant i < k$. Note that all $v_1, \ldots, v_k$ cannot point to any edge outside the cycle. Even when all edges are pointed to by both endpoints, by (ii), all $v_1, \ldots, v_k$ cannot point to any edge outside the cycle.

Assume towards contradiction that there are two undirected cycles C_1 and C_2 (not necessarily vertex or edge disjoint) in the same connected component of $\mathsf{Und}(\overrightarrow{G}\backslash S)$. If $V(C_1) \cap V(C_2) \neq \varnothing$, there must be a vertex $v \in C_2$ that points to an edge in $C_1 \backslash C_2$. This contradicts the above paragraph. If C_1 and C_2 are vertex disjoint, let $(v_1, \ldots, v_k)$ be an undirected path from C_1 and C_2 where $v_1 \in C_1$ and $v_k \in C_2$. By the above paragraph,

$\{v_1, v_2\}$ is uniquely pointed to by v_2 and inductively $\{v_i, v_{i+1}\}$ is uniquely pointed to by v_{i+1}. But applying the same argument from $\{v_{k-1}, v_k\}$, $\{v_i, v_{i+1}\}$ must be uniquely pointed to by v_i, leading to contradiction. Therefore, there must be only one undirected cycle in each connected component. ◀

After the rounding, each connected component has at most one cycle, so we can easily compute the optimal solution efficiently. Therefore, we compute a feasible solution that respects the constraints of the Feedback Vertex Set with Precedence Constraints. Since the total weights of deleted vertices in each step is at most LP/ε in Step (i), at most $2\mathsf{LP}/(\beta-\alpha)$ in Step (iii), and at most OPT in the final cleanup step, the final approximation ratio is

$$\frac{1}{\varepsilon} + \frac{2}{\beta-\alpha} + 1 \leqslant 62.2$$

by our choice of $\varepsilon = 0.0293258, \alpha = 0.514663, \beta = 0.588465$.

References

1. Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshtanov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In *LATIN 2016: theoretical informatics*, volume 9644 of *Lecture Notes in Comput. Sci.*, pages 1–13. Springer, Berlin, 2016. `doi:10.1007/978-3-662-49529-2_1`.
2. Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Polylogarithmic approximation algorithms for weighted-$\mathcal{F}$-deletion problems. In *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*, volume 116 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 1, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.
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):Art. 11, 28, 2019. `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*, pages 1711–1730. SIAM, Philadelphia, PA, 2019. `doi:10.1137/1.9781611975482.103`.
5. Jungho Ahn, Eduard Eiben, O-joung Kwon, and Sang-il Oum. A polynomial kernel for 3-leaf power deletion. In *45th International Symposium on Mathematical Foundations of Computer Science*, volume 170 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 5:1–5:14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020.
6. Andreas Brandstädt and Van Bang Le. Structure and linear time recognition of 3-leaf powers. *Inform. Process. Lett.*, 98(4):133–138, 2006. `doi:10.1016/j.ipl.2006.01.004`.
7. Yixin Cao. Linear recognition of almost interval graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1096–1115. ACM, New York, 2016. `doi:10.1137/1.9781611974331.ch77`.
8. Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(3):Art. 21, 35, 2015. `doi:10.1145/2629595`.
9. Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016. `doi:10.1007/s00453-015-0014-x`.
10. Chandra Chekuri and Vivek Madan. Constant factor approximation for subset feedback set problems via a new LP relaxation. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 808–820. ACM, New York, 2016. `doi:10.1137/1.9781611974331.ch58`.

11 Eduard Eiben, Robert Ganian, and O-joung Kwon. A single-exponential fixed-parameter algorithm for distance-hereditary vertex deletion. *J. Comput. System Sci.*, 97:121–146, 2018. doi:10.1016/j.jcss.2018.05.005.

12 Martin Farber. On diameters and radii of bridged graphs. *Discrete Math.*, 73(3):249–260, 1989. doi:10.1016/0012-365X(89)90268-9.

13 Samuel Fiorini, Gwenaël Joret, and Ugo Pietropaoli. Hitting diamonds and growing cacti. In *Integer programming and combinatorial optimization*, volume 6080 of *Lecture Notes in Comput. Sci.*, pages 191–204. Springer, Berlin, 2010. doi:10.1007/978-3-642-13036-6_15.

14 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-Deletion: approximation, kernelization and optimal FPT algorithms. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science—FOCS 2012*, pages 470–479. IEEE Computer Soc., Los Alamitos, CA, 2012.

15 Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michał Włodarczyk. Losing treewidth by separating subsets. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1731–1749. SIAM, Philadelphia, PA, 2019. doi:10.1137/1.9781611975482.104.

16 Pinar Heggernes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. *Theoret. Comput. Sci.*, 511:172–180, 2013. doi:10.1016/j.tcs.2012.03.013.

17 Bart M. P. Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. *SIAM J. Discrete Math.*, 32(3):2258–2301, 2018. doi:10.1137/17M112035X.

18 Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28(5):1906–1922, 1999. doi:10.1137/S0097539796303044.

19 Eun Jung Kim and O-joung Kwon. A polynomial kernel for distance-hereditary vertex deletion. In *Algorithms and data structures*, volume 10389 of *Lecture Notes in Comput. Sci.*, pages 509–520. Springer, Cham, 2017.

20 Eun Jung Kim and O-joung Kwon. Erdős-Pósa property of chordless cycles and its applications. *J. Combin. Theory Ser. B*, 145:65–112, 2020. doi:10.1016/j.jctb.2020.05.002.

21 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Trans. Algorithms*, 12(2):Art. 21, 41, 2016. doi:10.1145/2797140.

22 Stefan Kratsch and Magnus Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):Art. 20, 15, 2014. doi:10.1145/2635810.

23 Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, Geevarghese Philip, and Saket Saurabh. A $(2 + \epsilon)$-factor approximation algorithm for split vertex deletion. In *47th International Colloquium on Automata, Languages, and Programming*, volume 168 of *LIPIcs. Leibniz Int. Proc. Inform.* Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020.

24 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. doi:10.1007/s00453-008-9233-8.

25 Sang-il Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79–100, 2005. doi:10.1016/j.jctb.2005.03.003.

26 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

27 Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM J. Comput.*, 6(3):505–517, 1977. doi:10.1137/0206036.

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

29 Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Algebraic Discrete Methods*, 2(1):77–79, 1981. doi:10.1137/0602010.