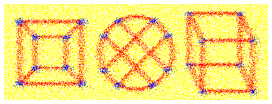
Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 18, no. 4, pp. 603–631 (2014)
DOI: 10.7155/jgaa.00339

Complexity of Disjoint Π-Vertex Deletion for Disconnected Forbidden Subgraphs

Jiong Guo Yash Raj Shrestha

Universität des Saarlandes,
Campus E 1.7, D-66123 Saarbrücken, Germany

Abstract

We investigate the computational complexity of Disjoint Π-Vertex Deletion. Here, given an input graph $G = (V, E)$ and a vertex set $S \subseteq V$, called a solution set, whose removal results in a graph satisfying a nontrivial, hereditary property Π, we are asked to find a solution set S' with $|S'| < |S|$ and $S' \cap S = \emptyset$. This problem is partially motivated by the "compression task" occurring in the iterative compression technique. The complexity of this problem has already been studied, with the restriction that Π is satisfied by a graph G iff Π is satisfied by each connected component of G [7]. In this work, we remove this restriction and show that, a few cases which are polynomial-time solvable, almost all other cases of Disjoint Π-Vertex Deletion are $\mathcal{NP}$-hard.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
March 2014	October 2014	October 2014	October 2014	December 2014

Published:
December 2014

Article type: Regular paper
Communicated by: S. Pal and K. Sadakane

Research supported by the DFG Excellence Cluster MMCI

E-mail addresses: jguo@mmci.uni-saarland.de (Jiong Guo) yashraj@mmci.uni-saarland.de (Yash Raj Shrestha)

1 Introduction

A *graph property* Π can be considered as a set of graphs. We say that a graph G *satisfies* Π if $G \in \Pi$. The classical Π-Vertex Deletion problem is defined as follows:

Π-Vertex Deletion (Π-VD)

Input: An undirected graph $G = (V, E)$ and a non-negative integer k.

Question: Is there a set S of at most k vertices whose removal results in a graph G' with $G' \in \Pi$?

Many prominent problems are special cases of Π-VD. For example, Vertex Cover is the case of Π being "edgeless". Lewis and Yannakakis [16] showed that Π-VD is $\mathcal{NP}$-complete for any non-trivial, hereditary property Π that can be verified in polynomial time. A graph property Π is *hereditary*, if it is closed under vertex deletion, and *non-trivial* if it is satisfied by infinitely many graphs and it is not satisfied by infinitely many graphs.

In the last 20 years, this $\mathcal{NP}$-completeness result motivated various research directions on Π-VD, for instance, approximation algorithms [1], its complexity on special input graphs [10], and the edge deletion counterpart [18]. One of the most remarkable recent approaches to cope with the $\mathcal{NP}$-completeness of Π-VD is parameterized algorithmics [19, 6]. Π-VD carries with its definition a natural parameter, the solution size k.

In 2004, Reed et al. [20] introduced the iterative compression technique [14, 19] which turned out to be particularly useful for achieving parameterized algorithms for Π-VD, for instance, for Undirected/Directed Feedback Vertex Set where Π is a set of all "acyclic' graphs' [4, 13, 2, 3] and Cluster Vertex Deletion where Π is a set of "disjoint union of cliques" [15]. This technique builds on two separate routines, namely, the iterative routine and the compression routine. In the former, we build the instance step by step from an empty instance, while in the latter we are given an instance and a solution, and we endeavor to search for a better solution for the given instance. The compression routines of iterative compression algorithms for Π-VD basically deal with a disjoint version of Π-VD which can be defined as follows:

Disjoint Π-Vertex Deletion (D-Π-VD) [7]

Input: An undirected graph $G = (V, E)$ and a vertex set $X \subseteq V$ such that $G[V \setminus X] \in \Pi$.

Question: Is there a vertex subset $X' \subseteq V$ with $|X'| < |X|$ and $X \cap X' = \emptyset$ such that $G[V \setminus X']$ satisfies Π?

Fellows et al. [7] initiated the study of D-Π-VD and gave a complexity dichotomy of this problem for the case that the non-trivial, hereditary property Π is determined by components. A graph property Π is determined by components, if a graph G satisfies Π iff each of G's connected components satisfies Π. Since every hereditary graph property Π can be characterized by a set $\mathcal{H}$

of forbidden induced subgraphs, the dichotomy achieved in [7] holds for D-Π-VD, where Π corresponds to a forbidden subgraph set $\mathcal{H}$ that contains only connected graphs. Fellows et al. [7] proved that as long as the forbidden set $\mathcal{H}$ does not contain a star with at most two leaves, the corresponding D-Π-VD problem is $\mathcal{NP}$-hard; otherwise, it is polynomial-time solvable.

In this paper, we generalize the results of Fellows et al. to the case where the forbidden set $\mathcal{H}$ is allowed to contain disconnected graphs. Note that many important graph properties can be characterized by forbidden sets $\mathcal{H}$ containing disconnected graphs, for instance, threshold graphs with $\mathcal{H}$ containing 4-vertex path, 4-vertex cycle and two independent edges. Recently, parameterized algorithms based on the iterative compression technique for Π-VD have been derived for Π corresponding to disconnected forbidden subgraph characterizations [12, 11].

Our Results Let $\mathcal{H}$ be the set of forbidden subgraphs corresponding to a graph property Π. Note that, if $\mathcal{H}$ or the set $\widetilde{\mathcal{H}}$ of the complement graphs of the graphs in $\mathcal{H}$ contains only connected graphs, the dichotomy in [7] applies. Thus, we only consider the case when both $\mathcal{H}$ and $\widetilde{\mathcal{H}}$ contain disconnected graphs. Our results can be summarized as follows: If $\mathcal{H}$ contain no star with at most two leaves and no disconnected graph whose connected components are all stars with at most two leaves, then D-Π-VD is $\mathcal{NP}$-complete. Then, for the case that $\mathcal{H}$ contains a star of two leaves, we prove polynomial-time solvability of D-Π-VD. For $\mathcal{H}$ containing disconnected forbidden subgraphs with stars of at most two leaves as connected components, we achieve $\mathcal{NP}$-hardness as well as polynomial-time solvability results. A few cases are left open.

Preliminaries. For a graph $G = (V, E)$, let $E(G)$ and $V(G)$ denote the set of edges and vertices of G, respectively. Unless specifically mentioned, we follow the graph theoretic notations and definitions from [5]. If we delete a vertex v or a subgraph S from graph G, we denote the resulting graph as $G - \{v\}$ and $G - S$, respectively. A *complement* or *inverse* of a graph G is a graph $\widetilde{G}$ such that $\widetilde{G}$ is on the same vertices as G and two vertices of $\widetilde{G}$ are adjacent if and only if they are not adjacent in G. For a set $\mathcal{H}$ of graphs, let $\widetilde{\mathcal{H}}$ be the set containing the complements of all graphs in $\mathcal{H}$.

A P_4 is a path on four vertices. A *star* S_l is a star with l leaves, while a $S_{\leq l}$ has at most l leaves. We call a graph a *non-star* if it does not satisfy the condition for stars. For a disconnected graph G, if each of its connected components is a star, we call G an *all-star*. A $(\geq i)$-all star is an all-star containing a star S_l with $l \geq i$. An (i)-all star is an all-star containing a star S_l with $l = i$. A $(\leq i)$-all star is an all-star which consists only of stars $S_{\leq i}$ as its connected components. Clearly, a (< 1)-all star is an edgeless graph. A $(P_4, 3)$-all star is a disconnected graph which consists only of P_4's and stars as its connected components, among which the largest star being S_3. A $(P_4, \leq 2)$-all star is a disconnected graph which consists only of P_4's and stars S_l with $l \leq 2$ as its connected components.

Given a set $\mathcal{H}$ of graphs, let $\mathcal{H}_c$ and $\mathcal{H}_d$ be the sets which contain all con-

nected and disconnected graphs of $\mathcal{H}$, respectively. Each graph $h_{di} \in \mathcal{H}_d$ can be viewed as a set of connected components, denoted as $h_{di(1)}, h_{di(2)}, \cdots, h_{di(t)}$. We say that $h_{di(j)}$ has a *number of occurrence* x in h_{di} if there exist x many connected components in h_{di} which are isomorphic to $h_{di(j)}$.

1.1 Some Important Definitions

A non-adjacent vertex of v is called a *non-neighbor* of v and the set of all non-neighbors of v is called the *non-neighborhood* of v. We say that v is *global* to a set Z if v is adjacent to all vertices of Z and we say that v is *non-adjacent* (or a *non-neighbor*) to a set Z if v is not adjacent to any vertex of Z. For two sets X and Y, we say X is global to Y when every vertex of X is adjacent to all vertices in Y and that X is *non-adjacent* to Y if there is no edge between X and Y.

Definition 1 *(Lexicographic order) Given sets A and B with a common ordering $\preceq$, one defines an ordering between all sequences (finite or infinite) of elements of A and of elements of B by $(a_1, a_2, \cdots) \preceq (b_1, b_2, \cdots)$ if either $a_i = b_i$ for every i, or $a_n < b_n$, where n is the first place in which they differ.*

Definition 2 *(α-sequence) [16] For a connected graph H, if H is 1-connected, then take a cut-vertex c; otherwise, let c be an arbitrary vertex (in this case, $H - \{c\}$ has only one connected component). Sorting the connected components of $H-\{c\}$ decreasingly with respect to their sizes gives a sequence $\alpha = (n_1, \cdots, n_i)$, where $n_1 \geq \cdots \geq n_i$. The sequence depends on the choice of c. The α-sequence of H, $\alpha(H)$, is a sequence which is lexicographically smallest among all such sequences α.*

Now, we present some definitions which will be used heavily in the reductions later. We classify the graphs in $\mathcal{H}$ and the connected components of graphs in $\mathcal{H}_d$ into one containing K_3 and K_3-free. Next, based on this classification we define the following set of lexicographic sequences and sets of connected components which will be crucial in the gadget constructions in the reductions that follows:

Definition 3 ($C(h_{di})$, $K(h_{di})$ and $J(h_{di})$) *For a disconnected graph h_{di}, the set $C(h_{di})$ consists of all the connected components of h_{di}. Similarly, the set $K(h_{di})$ consists of all the connected components of h_{di} which contains a K_3 and the set $J(h_{di})$ consists of all the connected components in $C(h_{di}) \setminus K(h_{di})$.*

Definition 4 (Ω-sequence, Γ-sequence and Δ-sequence) *For a disconnected graph h_{di}, the Ω-sequence of h_{di}, denoted as $\Omega(h_{di})$, is the lexicographically largest α-sequence of the α-sequences of all connected components of h_{di}. Similarly, the Γ-sequence of h_{di}, denoted as $\Gamma(h_{di})$, is the lexicographically largest α-sequence of the α-sequences of all connected components in $K(h_{di})$. If $K(h_{di})$ is empty, we set $\Gamma(h_{di})$ to $+\infty$. The Δ-sequence of h_{di}, denoted as $\Delta(h_{di})$ is the lexicographically largest α-sequence of the α-sequences of all connected components in $J(h_{di})$, if $K(h_{di})$ is empty; otherwise, we set $\Delta(h_{di})$ to $+\infty$.*

Definition 5 (Υ-sequence, Θ-sequence and Φ-sequence) *For a set of forbidden subgraphs $\mathcal{H}$, let A be the lexicographically smallest α-sequence of the α-sequences of all graphs in $\mathcal{H}_c$. Let B be the lexicographically smallest α-sequence of the α-sequences of all graphs in $\mathcal{H}_c$, which contains a K_3. Let C be the lexicographically smallest α-sequence of the α-sequences of all graphs in $\mathcal{H}_c$, which are K_3-free. Let D be the lexicographically smallest one of the Ω-sequences of all graphs in $\mathcal{H}_d$. Let E be the lexicographically smallest one of the Γ-sequences of all graphs in $\mathcal{H}_d$. Let F be the lexicographically smallest one of the Δ-sequences of all graphs in $\mathcal{H}_d$. The Υ-sequence of $\mathcal{H}$, denoted as $\Upsilon(\mathcal{H})$, is the lexicographically smaller one of A and D. The Θ-sequence of $\mathcal{H}$, denoted as $\Theta(\mathcal{H})$, is the lexicographically smaller one of B and E. The Φ-sequence of $\mathcal{H}$, denoted as $\Phi(\mathcal{H})$, is the lexicographically smaller one of C and F.*

Definition 6 ($\Psi(\mathcal{H})$ and $\Sigma(\mathcal{H})$) *For a set of graphs $\mathcal{H}$, the set $\Psi(\mathcal{H})$ consists of all the graphs h_{di} in $\mathcal{H}_d$, where h_{di} contains at least one connected component $h_{di(x)}$ such that $h_{di(x)}$ contains K_3 and $\alpha(h_{di(x)}) = \Theta(\mathcal{H})$. Similarly, the set $\Sigma(\mathcal{H})$ consists of all the graphs h_{dj} in $\mathcal{H}_d$, where h_{dj} is K_3-free and it has at least one connected component $h_{dj(y)}$ such that $\alpha(h_{dj(y)}) = \Phi(\mathcal{H})$.*

Definition 7 ($M(h_{di})$ and $N(h_{di})$) *For a graph $h_{di} \in \mathcal{H}_d$, the set $M(h_{di})$ consists of all the connected components $h_{di(x)}$ in h_{di}, such that $h_{di(x)}$ contains a K_3 and $\alpha(h_{di(x)}) = \Theta(\mathcal{H})$. Similarly, the set $N(h_{di})$ consists of all the connected components $h_{di(y)}$ in h_{di}, such that $h_{di(y)}$ is K_3-free and $\alpha(h_{di(y)}) = \Phi(\mathcal{H})$.*

Let ς be the number of leaves of smallest star $\mathcal{T}$ in $\mathcal{H}_c$. Let $\mathcal{U}$ be the all-star in $\mathcal{H}_d$ whose Ω-sequence is lexicographically largest among Ω-sequences of all all-stars in $\mathcal{H}$. Let τ be the number of leaves in largest connected component of $\mathcal{U}$.

2 No stars with at most two leaves

We prove here that, if $\mathcal{H}$ contain neither a star with at most two leaves nor an (≤ 2)-all-star, then D-Π-VD is $\mathcal{NP}$-hard. To this end, we distinguish two cases based on the number of leaves: First, we adapt the reduction in [7] to deal with the case that $\mathcal{H}$ contains no $S_{\leq 3}$ and no (≤ 3)-all stars. Then, a new reduction is given for the three leaf star case.

Lewis and Yannakakis [16] devised a framework, to prove that Π-VD for a non-trivial hereditary property Π is $\mathcal{NP}$-complete. Later, Fellows et al. [7] modified this framework for the $\mathcal{NP}$-hardness proofs of D-Π-VD with connected forbidden subgraphs when $\mathcal{H}$ contains no star with at most three leaves. They reduced from Vertex Cover on triangle-free graphs, where they picked the subgraph H in $\mathcal{H}$, that has the lexicographically smallest α-sequence[1], to build the vertex and edge gadgets [7]. We can further extend this adaptation to deal with disconnected forbidden subgraphs for the above case. However, the

[1]For the definition of α-sequences, see [16]

selection of the subgraph H for the vertex and edge gadgets is more tricky. Here, we have to consider the connected components of each disconnected subgraphs. If a connected component C of a disconnected subgraph H has lexicographically smallest α-sequence among all connected subgraphs in $\mathcal{H}_c$ and all connected components of disconnected subgraphs in $\mathcal{H}_d$, then we use C to build the vertex and edge gadgets. Further constructions are also needed for other components of H.

Before going to the reduction, we introduce a new variant of D-Π-VD, where the size of the new solution is given as a parameter. In all the following proofs, we show the $\mathcal{NP}$-hardness of this variant, since all hardness proofs need a *size gadget* that is employed in the reduction from the variant to D-Π-VD. We define this variant as follows:

Size Disjoint Π-Vertex Deletion (SD-Π-VD)

Input: An undirected graph $G = (V, E)$, an integer $k \geq 0$, and a vertex subset $X \subseteq V$ such that $G[X] \in \Pi$, $G[V \setminus X] \in \Pi$, and X is inclusion-minimal under the property Π.

Question: Is there a vertex subset $X' \subseteq V$ with $|X'| \leq k$ such that $X \cap X' = \emptyset$ and $G[V \setminus X'] \in \Pi$?

Lemma 1 *[7]* SD-Π-VD *can be reduced to* D-Π-VD *in polynomial-time.*

Note here that, the lemma and proof given in [7] for reduction from SD-Π-VD to D-Π-VD works even when $\mathcal{H}$ contains connected as well as disconnected graphs. In the following, we discuss the adaptation in the framework of Lewis and Yannakakis. This adaptation is similar to one in [7]; however, some modifications are needed to derive the general result. In each of the reductions, we will follow a chain of rules to choose an *encoding forbidden subgraph*, denoted by H and the *replacement structure*, denoted by $R(H)$. If the chosen encoding forbidden subgraph H is a connected forbidden graph, then we will take the whole H as our replacement structure $R(H)$; otherwise, if H is disconnected graph then we choose one of its connected components as our replacement structure $R(H)$. The process of choosing $R(H)$ is case-specific and will be discussed in details later.

Recall that, the α-sequence is defined based on a vertex c. Let $c \in R(H)$ be such a vertex whose removal results in the smallest α-sequence for $R(H)$. Note that every proper induced subgraph of $R(H)$ has a lexicographically smaller α-sequence than that of $R(H)$. When H is not a star, $R(H)$ must contain at least two vertices. Thus, a largest component J of $R(H) - \{c\}$ contains at least one vertex. Let d be an arbitrary vertex in J, and let $R(H)^\dagger$ be the graph resulting by removing all vertices in J from $R(H)$, and let J' be the subgraph of $R(H)$ induced by $V(J) \cup \{c\}$, as illustrated in Fig 1. If $H \in \mathcal{H}_d$, then we denote $H - R(H)$ as $S(G)$ and use $S(G)$ as *satellite gadget* which will be discussed in details later.

Recall that, for the following two proofs, we assume that the set $\mathcal{H}$ corresponding to Π contains no stars and all-stars. To proceed with the proof we

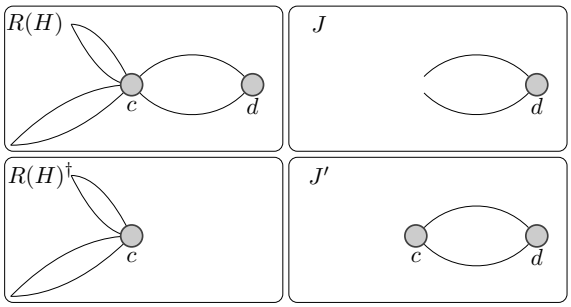

Figure 1: (from [7]) Connected graph $R(H)$ with a cut-vertex c and some other vertex d in a largest connected component J of $R(H) - \{c\}$. Moreover, the graphs $R(H)^\dagger := R(H) - V(J)$ and $J' = R(H)[V(J) \cup \{c\}]$ are illustrated.

make the following case distinction: 1) all graphs in $\mathcal{H}$ contains a K_3 and 2) not all graphs in $\mathcal{H}$ contains a K_3.

2.0.1 Case when all graphs in $\mathcal{H}$ contain a K_3:

Assume that $\mathcal{H}$ only consists of graphs that contains K_3. We reduce the $\mathcal{NP}$-complete Vertex Cover problem on K_3-free graphs [9] to Size D-Π-VD. Firstly, we follow some rules for choosing H from $\mathcal{H}$ and $R(H)$ from H.

1. If there is a connected graph H in $\mathcal{H}$ with the same α-sequence as $\Theta(\mathcal{H})$, then we use H as our encoding forbidden subgraph and also set $R(H) = H$.

2. Consider the case when there are no connected graphs in $\mathcal{H}$ with the same α-sequence as $\Theta(\mathcal{H})$, but there are disconnected graphs h_{di}'s $\in \mathcal{H}_d$ whose largest connected component has an α-sequence equal to $\Theta(\mathcal{H})$. Among the graphs in $\Psi(\mathcal{H})$, let $h_{di} \in \mathcal{H}_d$ be the one with minimum number of non-isomorphic connected components in their respective sets $M(h_{di})$. However, it may be possible that there exists another $h_{dk} \in \mathcal{H}_d$ such that all the non-isomorphic connected components $h_{di(j)}$'s in $M(h_{di})$ are also present in $M(h_{dk})$. We refer to all such h_{dk}'s as the K_3-*proxies* of h_{di}. Further, we make the following case distinction:

   (a) If there exists no K_3-proxy of h_{di} in $\mathcal{H}_d$, then we will choose h_{di} as H and set an arbitrary connected component from $M(h_{di})$ as our replacement structure $R(H)$.

   (b) If there exist K_3-proxies of h_{di} in $\mathcal{H}_d$, then we will pick a connected component $h_{di(j)}$ from $M(h_{di})$ such that h_{di} has the minimum number of occurrence of $h_{di(j)}$ among all its K_3-proxies. Moreover, we will choose h_{di} as H and set $h_{di(j)}$ as our replacement structure $R(H)$.

   (c) If there exists no such $h_{di(j)}$ in h_{di} which fulfils the condition given in (b), then we replace h_{di} with one of its K_3-proxies which contains

a connected component which fulfils such condition and we choose the necessary replacement structure as in (b).

Reduction and Correctness: In [7], Fellows et al. reduced an instance G of Vertex Cover on K_3-free graph to an instance G' of D-Π′-VD where the corresponding set of forbidden subgraphs $\mathcal{H}'$ contains no stars or disconnected subgraphs based on the framework of Yannakakis. In their reduction, the authors made heavy use of α-sequences. Firstly, we select the subgraph with lexicographically smallest α-sequence and mark it as encoding forbidden subgraph $R(H)$. For every vertex v in G create a copy of $R(H)^\dagger$ and identify c and v. Replace each edge $\{u, v\}$ in G with a copy of J', identifying c with u and d with v, respectively. Now, in the constructed graph G', we will greedily compute a minimal $\mathcal{H}$-obstruction set X for G' such that $X \cap V(G) = \emptyset$. Such a set X always exists, since G is K_3-free and, therefore, does not contain any forbidden subgraph. With this construction, the authors proved that that the graph G has a size-k vertex cover if and only if G' has a size-k vertex set disjoint from X that obstructs the graphs with same or higher α-sequence than that of $\alpha(R(H))$ in $G'[V \setminus V(S(G'))]$.

Now, in our reduction, we modify the adaptation of Yannakakis' reduction to include the scenario when both connected and disconnected graphs are in $\mathcal{H}$. Instead of choosing the subgraph with lexicographically smallest α-sequence, we follow the the rules above to choose the embedding forbidden subgraph $R(H)$ and create the D-Π-VD instance. We construct a set X similar to the one in [7]. Moreover if $H \in \mathcal{H}_d$, we initialize the satellite gadget of G' denoted as $S(G')$ with an isolated copy of $S(H)$ and add $V(S(G'))$ to X.

$R(H)$ contains a K_3. We can further observe that, if $H \in \mathcal{H}_c$, then all other graphs in $\mathcal{H}_c$ have the same or lexicographically larger α-sequences than the one of H and all graphs $h_{di} \in \mathcal{H}_d$ contain at least one connected component with lexicographically same or larger α-sequence than the one of H. If $H \in \mathcal{H}_d$ be the chosen encoding forbidden subgraph, it is evident from the rules followed that all the graphs in $\mathcal{H}_d$ have at least one connected component with α-sequence lexicographically equal to or larger than $\Theta(\mathcal{H})$. Moreover, all the graphs in $\mathcal{H}_d$ which are not K_3 proxies of H and contains no connected component with lexicographically higher α-sequence than $\Theta(\mathcal{H})$ contains at least one connected component h, such that $\alpha(h) = \Theta(\mathcal{H})$ and h is not present in H. Moverover, if the number of occurrence of $R(H)$ in H is x, then all h_{di}'s $\in \mathcal{H}_d$, which are K_3-proxies of H, contain a subgraph that is isomorphic to $R(H)$ and occurs in h_{di} at least x times. With these observations, it is evident that if the number of occurrence of $R(H)$ in H is x, we can obstruct all the graphs from $\mathcal{H}$ in G' by obstructing all but $x-1$ copies of $R(H)$ and all subgraphs with α-sequences higher than the one of $R(H)$ from G'.

With these observations and the arguments similar to the one in [7], we can show that the graph G has a size-k vertex cover if and only if G' has a size-k vertex set that obstructs all the graphs with same or higher α-sequence than that of $\alpha(R(H))$ in $G'[V \setminus V(S(G'))]$ which gives us the following lemma:

Lemma 2 *If $\mathcal{H}$ only consists of graphs which contains K_3, then* D-Π-VD *is $\mathcal{NP}$-hard.*

2.0.2 Case when not all forbidden subgraphs in $\mathcal{H}$ contain a K_3:

Similar to the above case, we follow some rules to choose the encoding forbidden subgraph H and the corresponding replacement structure $R(H)$ which will be used as a gadget in the reduction from $\mathcal{NP}$-complete Vertex Cover on K_3-free graphs to D-Π-VD. The only difference is that G might now contain a forbidden subgraph in $\mathcal{H}$, and we now need to show that it is possible to greedily compute a minimal $\mathcal{H}$-obstruction set X for G' such that $X \cap V(G) = \emptyset$. The rules are as follows:

1. If there is a K_3-free graph H in $\mathcal{H}_c$, with the same α-sequence as $\Phi(\mathcal{H})$, then we use H as our encoding forbidden subgraph H and also set it as $R(H)$.

2. Consider the case when there is no K_3-free graph in $\mathcal{H}_c$ with the same α-sequence as $\Phi(\mathcal{H})$, but there are other K_3-free graphs h_{di}'s $\in \mathcal{H}_d$ whose largest connected component has an α-sequence equal to $\Phi(\mathcal{H})$. Among the forbidden subgraphs in $\Sigma(\mathcal{H})$, let $h_{di} \in \mathcal{H}_d$ be the one with the minimum number of non-isomorphic connected components in their respective sets $N(h_{di})$. However, it may be possible that there exists another K_3-free $h_{dk} \in \mathcal{H}_d$ such that all the non-isomorphic connected components $h_{di(j)}$'s in $N(h_{di})$ are also present in $N(h_{dk})$, though their *number of occurrence* may be different. Again, we denote all such h_{dk}'s as the *K_3-free proxies* of h_{di}. Furthermore, we make the following case distinction:

   (a) If there exists no K_3-free proxy of h_{di} in $\mathcal{H}_d$, then we will choose h_{di} as H and any arbitrary connected component of h_{di} from $N(h_{di})$ as our replacement structure $R(H)$.

   (b) If there exist K_3-free proxies of h_{di} in $\mathcal{H}_d$, then we will pick the connected component $h_{di(j)}$ from $N(h_{di})$ such that h_{di} has the minimum number of occurrence of $h_{di(j)}$ among all its K_3-free proxies. Moreover, we will choose h_{di} as H and the connected component $h_{di(j)}$ as our replacement structure $R(H)$.

   (c) If there exists no such $h_{di(j)}$ in h_{di} which fulfils the condition given in (b), then we set H as one of the K_3-free proxies of h_{di}, which contains a connected component, which fulfils such conditions, and we choose the necessary replacement structure as in (b).

Reduction and Correctness: We construct G' in the same way as in the reduction in Lemma 2. Now, by setting $R(H)$ in this way, the largest connected component in $R(H) - \{c\}$ contains at least one edge, due to the fact that $R(H)$ is not a star. Moreover, since $R(H)$ does not contain K_3, at most one endpoint of this edge is adjacent to c. Then, we select an endpoint of this edge that is

not adjacent to c as the vertex d used in the construction of G'. Now, we can observe that in the resulting G' the vertices in $V(G)$ induce an independent set. Therefore, removing all vertices $V(G') \setminus V(G)$ gives an $\mathcal{H}$-obstruction set for G' and we can easily compute an inclusion-minimal solution X for G' with $X \cap V(G) = \emptyset$.

Since the graphs G and $R(H)$ are K_3-free and so is G', forbidden induced subgraphs with smaller α-sequences than $R(H)$, that contain K_3, do not need to be considered. We can further observe that, if $H \in \mathcal{H}_c$, then all other K_3-free graphs in $\mathcal{H}_c$ have same or larger α-sequences than the one of $R(H)$ and all the K_3-free graphs in $\mathcal{H}_d$ contain at least one connected component which has the same or larger α-sequence than the one of H. Moreover, if $H \in \mathcal{H}_d$, then all K_3-free graphs in $\mathcal{H}_c$ have α-sequences larger than the one of $R(H)$. If $H \in \mathcal{H}_d$ is the chosen encoding forbidden subgraph, it is evident from the rules followed that the connected component of H with the lexicographically largest α-sequence has an α-sequence equal to $\Phi(\mathcal{H})$. Moreover, all the K_3-free graphs in $\mathcal{H}_d$, which are not K_3-free proxies of H and contain no connected component with lexicographically higher α-sequence than $\Phi(\mathcal{H})$, contain at least one connected component h, such that $\alpha(h) = \Phi(\mathcal{H})$ and h is not present in H. If the number of occurrence of $R(H)$ in H is x, then all K_3-free forbidden subgraphs $h_{di} \in \mathcal{H}_d$, which are K_3-free proxies of H, contain a connected component that is isomorphic to $R(H)$, and occurs in h_{di} at least x times. From these observations, it is evident that if the number of occurrence of $R(H)$ in H is x, we can obstruct all the graphs from $\mathcal{H}$ in G' by obstructing all but $x-1$ copies of $R(H)$ and all K_3-free subgraphs with α-sequences higher than the one of $R(H)$ from G'.

With these observations and the arguments similar to the one in [7], we can show that the graph G has a size-k Vertex Cover if and only if G' has a size-k vertex set disjoint from X that obstructs all the K_3-free graphs with same or higher α-sequence than that of $\alpha(R(H))$ in $G'[V \setminus V(S(G'))]$ giving the following lemma:

Lemma 3 *If $\mathcal{H}$ has no stars and all-stars in $\mathcal{H}$, but contains other subgraphs such that there is at least one K_3-free subgraph in $\mathcal{H}$, then* D-Π-VD *is $\mathcal{NP}$-hard.*

Hence, from Lemmas 2 and 3, we get the Lemma 4

Lemma 4 D-Π-VD *is $\mathcal{NP}$-complete unless there is a star or an all-star in $\mathcal{H}$.*

2.1 Case with 4-stars and 4-all stars in $\mathcal{H}$

In this section, we present $\mathcal{NP}$-hardness proofs for D-Π-VD in the scenario when $\min\{\varsigma, \tau\} \geq 4$. Firstly, we follow the following rules to choose the encoding forbidden subgraph:

1. If $\varsigma \leq \tau$, we choose $\mathcal{T}$ as our encoding forbidden subgraph H.
2. If $\varsigma > \tau$, then among the all-stars in $\mathcal{H}_d$, we choose the one with the lexicographically smallest Ω-sequence as our encoding forbidden subgraph.

If more than one such all-stars have that particular Ω-sequence, then we choose the $h_{di} \in \mathcal{H}_d$ with the minimum number of occurrence of the connected component $h_{di(x)}$ such that $\alpha(h_{di(x)}) = \Omega(\mathcal{H})$ as our encoding forbidden subgraph H.

Clearly, the chosen encoding forbidden subgraph H is either a star or an all-star. Moreover, since there exists no (≤ 3)-all star in $\mathcal{H}$, if $H \in \mathcal{H}_d$, it contains at least one connected component $S_{\geq 4}$. If the chosen encoding forbidden subgraph H is a connected graph $h_{ci} \in \mathcal{H}_c$, we take the whole H as $R(H)$; otherwise, if H is a disconnected graph $h_{di} \in \mathcal{H}_d$, we choose its connected component with the largest α-sequence as $R(H)$ and use its other components as *satellite gadget*.

Based on the above observations, we now prove the following lemma:

Lemma 5 *If all the stars S_l present in $\mathcal{H}_c$ have $l \geq 4$ and all all-stars in $\mathcal{H}_d$ contain at least one connected component being a star S_l with $l \geq 4$, then* D-Π-VD *is $\mathcal{NP}$-hard.*

Proof: This proof is similar to the one in Section 4.3 of [7] with slight modifications to include the disconnected graphs in $\mathcal{H}$. Due to much similarity with the proof in [7], we just point out the modification to include the disconnected subgraphs in $\mathcal{H}$. In Lemma 4.13 in [7], Fellows et al. gave a reduction from an instance G of the $\mathcal{NP}$-complete Vertex Cover problem on graphs of maximum degree three [9] to an instance G' of SD-Π′-VD where the corresponding set of forbidden subgraphs $\mathcal{H}'$ contains stars with at least four leaves and no disconnected subgraphs. In the reduction, they constructed G' from G by embedding the smallest star in $\mathcal{H}'$ into every vertex and edge. In this way, they showed that G has a size-k Vertex Cover if and only if SD-Π′-VD has size-k solution. Now, in our adaptation, we construct the SD-Π-VD instance G' from Vertex Cover instance G by embedding the encoding forbidden subgraph $R(H)$ selected by the rules followed above into every edge and vertex of G instead of embedding the smallest star as in [7]. Let us assume $R(H)$ is a star with t leaves. Moreover, if H is a disconnected graph, we add $H - R(H)$ as the isolated satellite gadget $S(G')$ to G'. The vertices are added in X in the same way as in [7]. Moreover, we further add the entire satellite gadget to X.

Now, every graph in $\mathcal{H}_c$ has α-sequence lexicographically larger or equal to $\alpha(R(H))$. Moreover, every graph in $\mathcal{H}_d$ has at least one connected component with α-sequence lexicographically equal to or larger than the one of $R(H)$. Hence, it is enough to obstruct all non-stars and stars S_l with $l \geq t$ in $G'[V \setminus V(S(G'))]$ to satisfy the required property Π. With this observation and arguments similar to one in [7], we can show that there is a size-k vertex cover for G if and only if there is a vertex set X', $X' \cap X = \emptyset$, of size $k' := k + |E(G)|$, which obstructs all forbidden induced subgraphs in G'. In other words, (G, k) is a yes-instance for Vertex Cover if and only if (G', X, k') is a yes-instance for SD-Π-VD, which shows that SD-Π-VD is $\mathcal{NP}$-hard. The $\mathcal{NP}$-hardness of D-Π-VD then follows from Lemma 1. □

Hence, with Lemma 4 and Lemma 5 together we have the following lemma:

Lemma 6 *If $\mathcal{H}_c$ does not contain $S_{\leq 3}$ or (≤ 3)-all star, then* D-Π-VD *is $\mathcal{NP}$-hard.*

In the following, we consider the case that is, $\mathcal{H}$ contains S_3 or 3-all stars. Here, we distinguish two cases and derive completely different reductions compared to [7]. In particular, we prove the following theorem:

Lemma 7 *If there exists a star S_3 in $\mathcal{H}_c$ or an 3-all-star in $\mathcal{H}_d$ and there exists no $S_{\leq 2}$ or (≤ 2)-all star in $\mathcal{H}$, then* D-Π-VD *is $\mathcal{NP}$-hard.*

Proof: Here, we distinguish two cases: 1) $\mathcal{H}$ contains neither P_4 nor $(P_4, \leq 2)$-all star, and 2) $\mathcal{H}$ contains P_4 or $(P_4, \leq 2)$-all star. We firstly show the proof for case 1) and then the one for case 2) subsequently.

Assume that there exists a S_3 in $\mathcal{H}_c$ or a 3-all star in $\mathcal{H}_d$. Moreover, P_4 and $(P_4, \leq 2)$-all star are not present in $\mathcal{H}$. If there exists a S_3 in $\mathcal{H}_c$, we set H as S_3; otherwise, among the 3-all stars and $(P_4, 3)$-all stars in $\mathcal{H}_d$, we choose the one with the minimum number of occurrence of S_3 as H. Let this minimum number of occurrence be x. We can observe that all other graphs in $\mathcal{H}_d$ have either at least x occurrence of S_3 or it contains at least one connected component with a higher α-sequence than the one of S_3.

The reduction is from the $\mathcal{NP}$-complete 3SAT-2L problem [9], which is defined as follows:

3SAT-2l

Input: A 3-CNF boolean formula F where each literal appears at most twice in the clauses.

Output: A satisfying assignment for F.

We assume without loss of generality that each variable appears in each clause at most once. Let $F = c_1 \wedge \cdots \wedge c_q$ be a 3SAT-2L formula over a variable set $Y = \{y_1, \cdots, y_p\}$. We denote the k-th literal in clause c_j by l_j^k, for $1 \leq k \leq 3$. Starting with an empty graph G and $X := \emptyset$, construct an instance (G, X) for D-Π-VD as follows. An example of the construction is given in Fig 2. For each variable y_i, introduce a star Y_i with three leaves (variable gadget), add one leaf and the center vertex of Y_i to X and label the remaining leaves of Y_i with "+" and "−", respectively. For each clause c_j, add a star C_j with three leaves (clause gadget) and add its center vertex to X. Add a degree-1 neighbor to each vertex of C_j and this degree-1 vertex is added to X. Each of the three leaves of C_j corresponds to a literal in c_j, and each leaf is connected to a variable gadget as follows. Suppose that l_j^k is a literal of variable y_i, and let a_k be the leaf of C_j corresponding to l_j^k. Add an edge (connection gadget) between a_k and the "+"-leaf of the corresponding Y_i, if $l_j^k = y_i$; otherwise, to the "−"-leaf of Y_i. Finally, if H is a disconnected graph, a we create a *satellite gadget* $S(G)$ isomorphic to $H - S_3$, that is, the subgraph of H with one S_3 removed. Moreover, we add all vertices of this satellite gadget to X.

Obviously, $G[V \setminus X]$ only contains disjoint stars with at most two leaves. We note here that, since each literal can occur in at most two clauses, these two

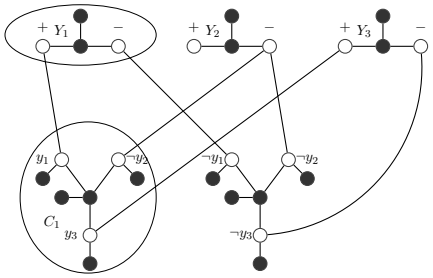

Figure 2: Example for the reduction in the proof of Theorem 7 from the 3SAT-2L instance with formula $(y_1 \wedge \neg y_2 \wedge y_3) \vee (\neg y_1 \wedge \neg y_2 \wedge \neg y_3)$ to Disjoint Π-Vertex Deletion with the given solution X (black vertices). For illustration the clause gadget C_1 and variable gadget Y_1 are labelled.

leaf stars can only be centered at "+" or "−" labelled vertices of Y_i. Hence, $G[V \setminus X] \in \Pi$. We now show that formula F has a satisfying truth assignment if and only if there exists a size-$(p+2q)$ set X' with $X' \cap X = \emptyset$, that obstructs all forbidden induced subgraphs in G. Clearly, $|X| > p + 2q$.

($\Rightarrow$) Assume that a satisfying truth assignment for F is given. Based on this truth assignment, we construct the disjoint solution X', beginning with $X' := \emptyset$, as follows. For each variable y_i, $1 \leq i \leq p$, if y_i = TRUE, then add the vertex labelled "+" in Y_i to X'; otherwise, add the vertex labelled "−" to X'. This ensures that from each variable gadget, exactly one vertex will be in X'. Next, for each clause c_j we have at least one literal set TRUE, say l^i_j. Then, we add the two leaves of C_j which do not correspond to l^i_j to X'. This procedure obstructs the stars with three or more leaves at the clause gadgets by totally $2q$ vertices. Then, $|X'| = p + 2q$. The connected components of $G[V \setminus (X' \cup V(S(G)))]$ are either isolated vertices, isolated edges, or P_4's. Hence, $G[V \setminus X'] \in \Pi$.

($\Leftarrow$) Let X' with $X' \cap X = \emptyset$ be a size-$(p+2q)$ vertex set that obstructs all forbidden induced subgraphs in G. Due to the satellite gadget $S(G)$, X' must obstruct all S_3's in $G[V \setminus V(S(G))]$. Since there exists a S_3 in each variable gadget, at least one vertex from each variable gadget must be in X', which requires in total p vertices. This means that we can construct an assignment for F: From a variable gadget Y_i, if the vertex labelled "+" is in X', we assign TRUE to y_i; otherwise, we assign false to y_i. The other $2q$ vertices of X' must be used for obstructing the stars with three or more leaves at clause gadgets, which implies that for each clause gadget c_j, exactly one leaf remains in $G[V \setminus X']$. The constructed assignment has to satisfy the formula F, since for each connection gadget, there exist some S_3's, each consisting of this connection gadget and a vertex in X and thus, from each connection gadget at least one vertex is in X'. □

Now, we prove Lemma 7 for the case 2) that is, when $\mathcal{H}$ contains P_4 or $(P_4, \leq 2)$-all star.

Proof:

Assume that there exists a P_4 or a $(P_4, \leq 2)$-all star in $\mathcal{H}$. We make further case distinctions:

A **There exists a P_4 in $\mathcal{H}$.**

In this case, if there exists a S_3 in $\mathcal{H}_c$, set $H_1 = S_3$ and $R(H_1) = S_3$ and set $s = 1$; otherwise among the 3-all stars in $\mathcal{H}$, we pick the one with the minimum number of occurrence of S_3 as encoding forbidden subgraph H_1 and set $R(H_1) = S_3$. Set s with the number of occurrence of S_3 in H_1. Moreover, we set $H_2 = P_4$ and $t = 1$.

B **There does not exist a P_4 in $\mathcal{H}$.**

Among $(P_4, \leq 2)$-all stars in $\mathcal{H}$, let T be the one with the minimum number of occurrence of P_4. We set $H_2 = T$ and $R(H_2) = P_4$ and initialize $S(G)$ with an isolated copy of $H_2 - V(R(H_2))$. Let the number of occurrence of P_4 in T be t. If there exists a S_3 in $\mathcal{H}_c$, set $H_1 = S_3$ and $s = 1$; otherwise, among the 3-all stars and $(P_4, 3)$-all stars with at most t occurrence of P_4's, we choose the one with the minimum number of occurrence of S_3 and set it as H_1. In this case, let the number of occurrence of star S_3 in H_1 be s. Based on H_1, we make the following case distinctions:

(a) H_1 **is a 3-all star or a S_3.**
In this case, set $R(H_1) = S_3$. Moreover, add $S(G)$ with an isolated copy of $H_1 - V(R(H_1))$.

(b) H_1 **is a $(P_4, 3)$-all star.**
In this case, set $R(H_1) = S_3$. Moreover, add an isolated copy of $H_1 - V(R(H_1))$ in satellite gadget $S(G)$. Further, we remove all but $t - 1$ P_4's from $S(G)$.

Note here that if all the $(P_4, 3)$-all stars in $\mathcal{H}$ contain higher occurrence of P_4 than T, then we can remove all $(P_4, 3)$-all stars from $\mathcal{H}$ and do the reduction as in Lemma 1. This is true as in the reduction, T will take care of all the removed $(P_4, 3)$-all stars.

We assume that there exists a S_3 in $\mathcal{H}_c$ or there exists a 3-all star in $\mathcal{H}_d$ and there also exists a P_4 or a $(P_4, \leq 2)$-all star in $\mathcal{H}$. Let H_1 and H_2 be the corresponding graphs and $S(G)$ the corresponding satellite gadget chosen based on the rules followed above. The reduction from $\mathcal{NP}$-complete 3SAT-2L is similar to one for 1). Starting with an empty graph G and $X := \emptyset$, construct an instance (G, X) for D-Π-VD as follows. For each variable y_i, introduce a path Y_i on five vertices (variable gadget), and add the end and middle vertices on each Y_i to X and label the remaining vertices of Y_i with "+" and "−", respectively. For each clause c_j, add a star C_j with four leaves (clause gadget) and add its central vertex and one of its leaves to X. Each of the three remaining leaves of C_j corresponds to a literal in c_j, and each leaf is connected to a variable gadget as follows. Suppose that l_j^k is a literal y_i or $\neg y_i$, and let a_k be the leaf

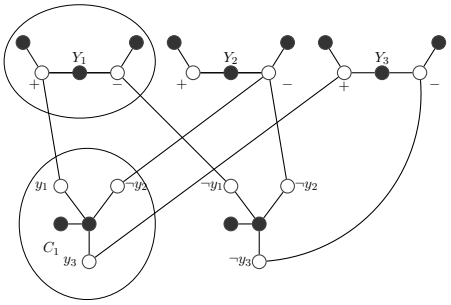

Figure 3: Example for the reduction in the proof of case 2) of Lemma 2 from the 3SAT-2L
instance with formula $(y_1 \wedge \neg y_2 \wedge y_3) \vee (\neg y_1 \wedge \neg y_2 \wedge \neg y_3)$ to D-Π-VD with the given solution X (black vertices). For illustration the the clause gadget C_1 and variable gadget Y_1 are labelled.

of C_j corresponding to l_j^k. Add an edge (connection gadget) between a_k and "+" of the corresponding Y_i if a_k corresponds to positive literal; otherwise, to "−" of the corresponding Y_i. Finally, we add an isolated copy of satellite gadget $S(G)$ to G and we add the satellite gadget to X. An example of the following construction is given in Fig. 3.

Observe that every forbidden subgraph in $\mathcal{H}$ contains at least one connected component which is S_3 or a P_4 or one with higher α-sequence than that of a star S_3. Obviously, $G[V \setminus X]$ only contains disjoint stars with at most two leaves. We note here that since each literal can occur in at most two clauses, there can be a star with at most two leaves at "+" or "−" labelled vertex of Y_i. Hence, $G[V \setminus X] \in \Pi$. With the above construction, and arguments similar to the proof of 1), we can show that formula F has a satisfying truth assignment if and only if there exists a size-$(p + 2q)$ set X', $X' \cap X = \emptyset$, that obstructs all P_4's and S_3's in $G'[V \setminus V(S(G'))]$. In other words, F has a satisfying truth assignment if and only if $(G, X, p + 2q)$ is a yes-instance of D-Π-VD. □

Combining Lemmas 6 and 7, we arrive at the following theorem:

Theorem 1 *D-Π-VD is $\mathcal{NP}$-complete, unless $\mathcal{H}$ contains a star $S_{\leq 2}$ or a (≤ 2)-all star.*

3 Stars with two leaves

We examine now the cases that $\mathcal{H}$ contains $S_{\leq 2}$ or (≤ 2)-all star. Here, we achieve $\mathcal{NP}$-completeness as well as polynomial-time solvability results. First we show that the case with $S_{\leq 2} \in \mathcal{H}$ is solvable in polynomial time.

3.1 Forbidden stars with two leaves

Fellows et al. [7] proved that in the case of connected forbidden subgraphs, if $\mathcal{H}$ contains $S_{\leq 2}$, then D-Π-VD is polynomial-time solvable. We extend this result to the disconnected case.

Theorem 2 D-Π-VD *can be solved in polynomial time when* $\mathcal{H}$ *contains a star* $S_{\leq 2}$.

This theorem applies to the graph properties Π whose sets of minimal forbidden induced subgraphs $\mathcal{H}$ contain K_1 (a single vertex), P_2 (a single edge) or P_3 (a path on three vertices). D-Π-VD with K_1 being forbidden is not non-trivial, as Π is an empty set, and hence solvable in polynomial time. If $\mathcal{H}$ contains P_2, then other minimal forbidden subgraphs in $\mathcal{H}$ can only be sets of independent vertices; otherwise, these forbidden subgraphs are not minimal. If $\mathcal{H}$ consists of only P_2, then the compression routine in the iterative compression algorithm for Vertex Cover given in [19] directly gives a polynomial-time algorithm for the corresponding D-Π-VD problem. If $\mathcal{H}$ contains in addition to P_2, an independent set of size s, then Π is not non-trivial. Now, we give a polynomial-time algorithm for D-Π-VD when $\mathcal{H}$ contains P_3.

Note that a graph is called a *cluster graph* if it contains no induced P_3. A cluster graph is a disjoint set of cliques. Since $\mathcal{H}$ is a set of minimal forbidden induced subgraphs and $P_3 \in \mathcal{H}$, all other forbidden subgraphs in $\mathcal{H}$ must also be cluster graphs. Let $H_1, H_2, \ldots, H_l$ be the minimal cluster graphs present in $\mathcal{H}$. Now, for each forbidden cluster graph H_i, let $H_i^1, H_i^2, \ldots, H_i^c$ be its connected components arranged in non-ascending order of their sizes. Let $(G = (V, E), X)$ be the input instance of D-Π-VD with $|V| = n$. Since X is a solution set, $G[X]$ is a collection of cliques and it induces no forbidden cluster graphs from $\mathcal{H}$.

We describe an algorithm that finds a minimum-size vertex set X' such that $X \cap X' = \emptyset$ and $G[V \setminus X'] \in \Pi$, or returns "no-instance". The algorithm is similar to the compression routine of the iterative compression algorithm for Cluster Vertex Deletion [15], but additionally takes into account the forbidden cluster graphs in $\mathcal{H}$. In first step, the instance is simplified by two simple data reduction rules, whose correctness is easy to see [15]:

1. Delete all vertices in $R := V \setminus X$ that are adjacent to more than one clique in $G[X]$.

2. Delete all vertices in R that are adjacent to some, but not all vertices of a clique in $G[X]$.

After these data reduction rules have been exhaustively applied, the instance has the following property. In each clique of $G[R]$, we can divide the vertices into equivalence classes according to their neighborhoods in X, where each class then contains vertices either adjacent to all vertices of a particular clique in $G[X]$, or adjacent to no vertex in X. This classification is useful because of the following:

Lemma 8 *[14] If there exists a solution for D-Π-VD, then in the cluster graph resulting by this solution, for each clique in $G[R]$ the vertices of at most one equivalence class are present.*

Due to Lemma 8, the remaining task for solving D-Π-VD is to assign each clique in $G[R]$ to one of its equivalence classes in such a way that the forbidden cluster graphs in $\mathcal{H}$ are also obstructed. Hence, in our algorithm we will enumerate all the possiblities which will not induce any forbidden cluster graph and choose the one with the minimum number of vertex deletions. However, we cannot do this independently for each clique in $G[R]$. The reason is that we cannot choose two classes from different cliques in $G[R]$ that are adjacent to the same clique in $G[X]$, since this would create an induced P_3. This assignment problem can be modelled as a *weighted bipartite matching* problem in an auxiliary graph $J = (V', E')$ where each edge corresponds to a possible choice of a clique. Let $|E'| = m$. Moreover, for each edge $e \in E'$, we set two weights $w_1(e)$ and $w_2(e)$ which will be instrumental while enumerating the possibilities. We delete a set X'_1 of vertices while eumerating the possibilities and create another set X'_2 by the weighted bipartite matching procedure which will be explained later. Among all the resulting graphs which satisfy Π, we choose the one with the minimum cardinality of $X' = X'_1 \cup X'_2$. The graph J is constructed as follows. See Fig 4 for an illustration.

1. For every clique in $G[R]$ which has at least one neighbour in $G[X]$, add a vertex (white vertex) in J.

2. For every clique C_X in $G[X]$ which has at least one neighbour in $G[R]$, add a vertex v (black vertex in X) in J. Moreover, add a new degree-1 vertex u (white vertex in X) and an edge $\{u, v\}$. Set the weights w_1 and w_2 of this edge $\{u, v\}$ to be the size of C_X. This edge corresponds to choosing C_X and removing all vertices adjacent to C_X from $G[R]$.

3. For a clique C_X in $G[X]$ and a clique C_R in $G[R]$, add an edge e between the vertex for C_X and the vertex for C_R if there is an equivalence class in C_R adjacent to C_X. This edge corresponds to choosing this class for C_R and is assigned two different weights, $w_1(e)$ is assigned the total number of vertices in the corresponding class of C_R and in C_X and $w_2(e)$ is assigned the total number of vertices in C_R.

4. Add a vertex for the class in a clique C_R that is not adjacent to any clique in $G[X]$ (black vertices outside X), and connect it to the vertex representing C_R. Again, this edge corresponds to choosing this class for C_R and both its weights w_1 and w_2 are assigned the total number of vertices in this class.

5. For each clique C_R in $G[R]$ which is non-adjacent to any vertex in $G[X]$, add an edge e between two new grey vertices and its weights $w_1(e)$ and $w_2(e)$ are set to the total number of vertices in C_R.

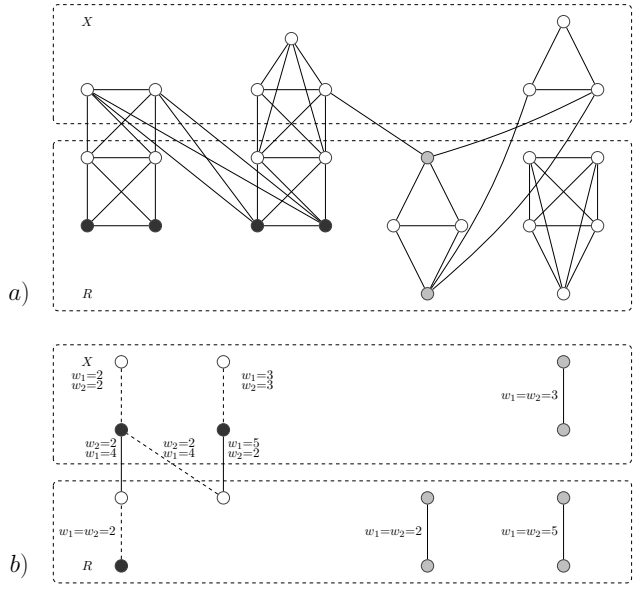

Figure 4: a): Data reduction in the algorithm for D-Π-VD when P_3 is present in $\mathcal{H}$. The grey vertices are deleted by the data reduction rules. The black vertices correspond to the minimal solution X_2' determined by the solution of weighted bipartite matching problem. b) The graph represents the corresponding weighted bipartite matching instance with the edge weights represented by integers next to the edges. The dashed edges represent the cliques that are added into X_2' while the black edges represent the remaining cluster graph.

6. For each clique C_X in $G[X]$ which is non-adjacent to any vertex in $G[R]$, add an edge e between two new grey vertices and its weights $w_1(e)$ and $w_2(e)$ are equal to the total number of vertices in C_X. We denote the set of these edges by T.

Since, we only added edges between black and white vertices and isolated edges between two grey vertices, J is bipartite. The task is now to find a *maximum-weight bipartite matching* on J with edge weights w_2, that is a set of edges of maximum weight where no two edges have an endpoint in common. However, before that we must take care that the collection of cliques resulting from the matchings does not induce any forbidden cluster graph in $\mathcal{H}$. To this end, we enumerate all matchings in graph $G[V \setminus X']$ which do not induce any forbidden cluster graphs from $\mathcal{H}$ with the help of the edge weights w_1.

We use *cluster configurations* and the corresponding *cluster configuration diagrams* for enumerating all such matchings. An example is illustrated in Fig 5 for the case that $\mathcal{H}$ consists of P_3 and a cluster graph h_1 with four cliques h_1^1, h_1^2, h_1^3 and h_1^4 of size $s_1 \geq s_2 \geq s_3 \geq s_4$, respectively. The cluster configuration

diagram is a table in which columns represent the *size intervals* of the allowable cliques. For example, in Fig 5, the column between h_1^1 and h_1^2 represents the size interval of $[s_2, s_1)$. Each row depicts a feasible configuration. Here, "↑" represents a clique in the corresponding interval and "⇛"'s position represents the upper bound on the size of the remaining cliques.

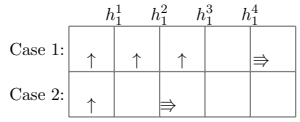

Figure 5: Cluster configuration diagram when the corresponding forbidden subgraphs in $\mathcal{H}$ consists of a P_3 and a cluster graph h_1 with four cliques as connected components.

Lemma 9 *The number of different feasible clique configurations of $G[V \setminus X']$ in the case that $\mathcal{H}$ consists of P_3 and cluster graphs, is polynomial in the size of G.*

Proof: Let the number of forbidden cluster graphs in $\mathcal{H}$ be d. Let b be the number of disjoint cliques in the forbidden cluster graph in $\mathcal{H}$, which has the maximum number of disjoint cliques among all cluster graphs in $\mathcal{H}$. Then, in the *cluster configuration diagram* there will be at most $bd + 1$ intervals, since the number of distinct-sized cliques, which are connected components of cluster graphs in $\mathcal{H}$, is at most bd. Furthermore, we observe that for any fixed position of "⇛", say s, there can be at most $b - 1$ cliques represented by entries with "↑" to the left of s. Since each forbidden cluster graph in $\mathcal{H}$ has at most b components, any cluster configuration Y with more than $b-1$ cliques to the left of s will be handled by a different cluster configuration Z, such that Z is formend from Y by removing the rightmost "↑"and "⇛" is moved to the beginning of the interval containing the removed "↑". Hence, for each fixed position s for "⇛", from all possible cliques with size at least s, we need to pick at most $b-1$ entries to add the "↑"'s to the left of s. The number of the different ways of arranging $b-1$ ↑'s in $bd+1$ intervals is bounded by $(b-1)^{(bd+1)}$. Now, for each such arrangement, we can pick each ↑ from the edges in the reduced graph J. We note here that every edge with weight w_1 that does not fit into a particular interval can be trimmed down by adding some vertices of the corresponding clique to X_1'. This ends up with $m^{(b-1)^{(bd+1)}}$ possibilities. Let $c = (b-1)^{(bd+1)}$. Now, since for each fixed Π, $\mathcal{H}$ is fixed, hence, b and d are constants, i.e., c is a constant. Since there are at most $(bd + 1)$ positions to fix "⇛", the total number of configurations is a polynomial function of m, i.e., $\mathcal{O}(m^c)$. □

From all cluster configurations, we choose only those configurations which are feasible for a set of forbidden subgraphs $\mathcal{H}$, which can be done in polynomial time. Any feasible configuration places the cliques in $G[X]$ with sizes greater

than the position of "⇛" at the positions set by "↑". Moreover, in any feasible configuration, if any edge from J is selected to fill the position for "↑", none of its adjacent edges can be picked to fill the positions of other "↑" in the same configuration. Moreover, it also places all cliques represented by edges in T in their corresponding intervals. Next, for each feasible configuration we do the following:

1. We remove the vertices at the end-points of the edges corresponding to the chosen cliques denoted by "↑".
2. In the remaining graph J, we maintain an upper-bound corresponding to the position of "⇛" on the weights w_1 of the remaining edges. Let the position of the "⇛" be s. For each edge c corresponding to clique C with $w_1(c) \geq s$, add $|V(C)| - s$ arbitrary vertices from $V(C) \cap R$ to X_1' and reduce the weights $w_1(c)$ and $w_2(c)$ by $w_1(c) - s$.
3. Now, we run the algorithm for maximum weighted bipartite matching on the remaining graph with weights w_2. Since the weight of each edge in J is bounded by n, the size of the given instance, we get a running time of $\mathcal{O}(m^4\sqrt{n}\log n)$ [8].

The set X_2' can be directly constructed from a maximum matching returned in Step 3; it contains all vertices in the equivalence classes in $G[R]$ that correspond to the edges not chosen by the matching. Hence, our disjoint solution is $X' = X_1' \cup X_2'$. Clearly, both X_1' and X_2' can be computed in polynomial time. Combined with the polynomial number of configurations we have an overall polynomial-time algorithm.

3.2 All-stars containing stars of at most two leaves

For this case, we cannot give a complete dichotomy of the complexity of D-Π-VD. However, we can present some $\mathcal{NP}$-hard and polynomial cases. For example, we have the following result for 2-all star.

Lemma 10 *If there exists 2-all star in $\mathcal{H}_d$, each of which contains at least two occurrence of star S_2 as their connected components, and $\mathcal{H}$ is free from the following graphs, then* D-Π-VD *is $\mathcal{NP}$-hard:*

1. *A star $S_{\leq 2}$.*
2. *A (≤ 2)-all star with only one occurrence of S_2 as its connected component.*
3. *A (< 2)-all star.*
4. *A subgraph which is an induced subgraph of a graph $G' = (V', E')$ with the following properties:*
    i. *There exists a set $M \subset V'$ which induces a clique.*

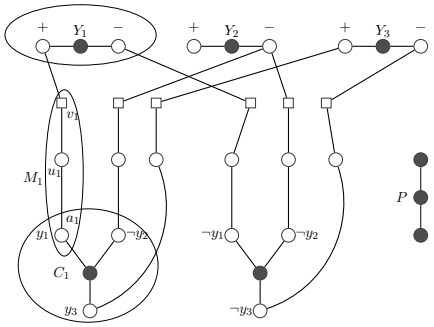

Figure 6: Example for the reduction in the proof of Lemma 10 from the 1-IN-3 SAT
instance with formula $(y_1 \wedge \neg y_2 \wedge y_3) \vee (\neg y_1 \wedge \neg y_2 \wedge \neg y_3)$ to D-Π-VD with the given solution X (black vertices). The vertices in clique M is illustrated by square boxes. For illustration the the clause gadget C_1, connection gadget M_1 and variable gadget Y_1 are labelled.

ii. For each vertex $v \in M$, there exists an edge $\{x, y\}$, such that either $\{x, v\} \in E'$ or $\{y, v\} \in E'$ and $N(x) \cup N(y) = \{x, y, v\}$. Let $N \subset V'$ be the set which contains all such vertices $\{x, y\}$ for each $v \in V'$.

iii. $J := V' \setminus (M \cup N)$ induces an independent set in G' such that for every $v \in J$, $N(v) \subseteq M$.

5. A disconnected forbidden graph which consists of graphs in (1-4) as its connected components.

Proof:

We first show how D-Π-VD is $\mathcal{NP}$-hard when one of the graphs in $\mathcal{H}_d$ is a disjoint union of two P_3's, i.e., $2P_3$. Later, we show how this reduction can be adapted for the general case. The proof is by reduction from 1-IN-3 SAT to SD-Π-VD. We define 1-IN-3 SAT as follows:

1-in-3 Sat

Input: 3-CNF boolean formula F.

Output: A truth assignment which makes exactly one literal in each clause TRUE, or a certificate that no such assignment exists.

1-IN-3 SAT is $\mathcal{NP}$-complete [21]. We assume without loss of generality that each variable appears in each clause at most once. Let $F = c_1 \wedge \cdots \wedge c_q$ be a 3-CNF formula over a variable set $Y = \{y_1, \cdots, y_p\}$. We denote the k-th literal in clause c_j by l_j^k, for $1 \leq k \leq 3$. An example of the following construction is given in Fig. 6. Starting with an empty graph G and $X := \emptyset$, construct an instance (G, X) for D-Π-VD as follows. For each variable y_i, introduce a

path Y_i on three vertices (variable gadget), add the middle vertex on Y_i to X and label the remaining vertices of Y_i with "+" and "−" respectively. For each clause c_j, add a star C_j with three leaves (clause gadget) and add its center vertex to X. Each of the three leaves of C_j corresponds to a literal in c_j, and each leaf is connected to a variable gadget as follows. Suppose that l_j^k is a literal y_i or $\neg y_i$, and let a_t be the leaf of C_j corresponding to l_j^k. For each such leaf a_t of the clause gadget add a path M_t (connection gadget) of three vertices, identify one of its end vertices with a_t. Among the remaining vertices in M_t, we denote the middle vertex as u_t and the other end vertex as v_t. We add an edge between v_t and the vertex of Y_i labelled "+", if l_j^k is positive literal; otherwise with the one labelled "−". Add all possible edges between the v_t's corresponding to connection gadgets. Finally, add a path P of three vertices disjoint from the preceding construction to G and add all its vertices to X.

Obviously, $G[V \setminus X]$ only contains a clique M which consists of the vertices v_t's of connection gadgets M_t's with a P_2 attached to each vertex of M and a set of vertices $J \subset V$ which induces an independent set in G. Moreover, for each $v \in J$, $N(v) \subseteq V(M)$ (refer to Figure 6). Hence, all the induced P_3's contain at least one of its vertices in the clique M, and $G[V \setminus X] \in \Pi$. We show that formula F has a satisfying truth assignment if and only if there exists a size-$(p + 5q)$ set X', $X' \cap X = \emptyset$, that obstructs all the forbidden induced subgraphs in G. In other words, F has a satisfying truth assignment if and only if $(G, X, p + 5q)$ is a yes-instance of SD-Π-VD. The $\mathcal{NP}$-hardness of D-Π-VD then follows from Lemma 1.

($\Rightarrow$) Assume that a satisfying truth assignment for F is given. Based on this truth assignment, we construct the disjoint solution X', beginning with $X' := \emptyset$, as follows: For each variable y_i, $1 \leq i \leq p$, if y_i = TRUE, then add the vertex labelled "+" in Y_i to X' and the vertices corresponding to $\neg y_i$ in the clause gadgets to X'. Moreover, we also add the u_t's corresponding to all M_t's connected to the "+" vertex of Y_i to X'. If y_i = FALSE, we add the vertex labelled "−" in Y_i and the vertices corresponding to y_i in the clause gadgets to X'. Moreover, we also add the v_t corresponding to all M_t's connected to the "−" vertex of Y_i to X'. We observe that from each variable gadget we take exactly one vertex in X', resulting totally in p vertices deletion. From each connection gadget, we keep either u_t or v_t in X'; resulting totally in $3q$ vertices deletion. Finally, from each clause gadget, we take exactly two vertices corresponding to FALSE literals in X' resulting totally in another $2q$ vertices deletion. Clearly, $|X'| = p + 5q$. The connected components of $G[V \setminus X']$ are either isolated vertices or paths on at most two vertices, an isolated path P on three vertices and an isolated clique comprising of remaining v_t's of M_t's which could not be included in X' (thus $G[V \setminus X'] \in \Pi$).

($\Leftarrow$) Let X', $X' \cap X = \emptyset$, be a size-$(p + 5q)$ vertex set that obstructs every forbidden induced subgraph in G. Recall that the set of minimal forbidden induced subgraphs contains a $2P_3$. Since there exists a path P on three vertices in X which is disjoint from every other vertex in $G[V \setminus V(P)]$, it is necessary that $G[V \setminus (X' \cup V(P))]$ should be free from P_3. Now, since each variable

gadget Y_i comprises of an induced P_3, at least one vertex from each Y_i must be in X'. This results in a total of at least p vertices in X'. Now, since each clause gadget C_i is a S_3, at least 2 leaves from each C_i must be in X'. This results in a total of at least $2q$ vertices in X'. Since v_t's form a clique, in each connection gadget M_t, either v_t or u_t needs to be in X' to obstruct induced P_3's. We can observe here that keeping all u_t's or v_t's in X' does not give a better solution. If we keep all u_t's in X', there may exist some P_3's formed by v_t's and remaining vertices of vertex gadget. Similarly, if we keep all v_t's in X', there may exist some induced P_3's formed by u_t's and remaining vertices of clause gadget. Hence, for each clause at least three vertices among u_t's and v_t's corresponding to its literals must be in X' which results in a total of at least $3q$ vertices in X'. We can observe here that the only way to obstruct P_3's in $G[V \setminus (X' \cup V(P))]$ with at most $p+5q$ vertices is to take alternating vertices in the paths formed by a_t, u_t, v_t and the vertices in variable gadget. From a vertex gadget Y_i, if the vertex labelled "+" is in X', we assign TRUE to variable y_i. On the other hand, if the vertex labelled "−" is in X', we assign FALSE to variable y_i. This assignment has to satisfy the formula since in every clause exactly one literal is true by our gadget's construction. $G[V \setminus (X' \cup V(P))]$ only comprises of an isolated clique and disconnected stars S_l with $l < 2$ as its connected components (thus $G[V \setminus X'] \in \Pi$).

To this end, we show how this proof can be adapted for the general case. Among the (≤ 2)-all star from $\mathcal{H}$, we pick the one with the minimum number of occurrence of P_3, let that forbidden subgraph be H. Let the number of occurrence of P_3 in H be x, where $x \geq 2$. We set one of the P_3's contained in H as $R(H)$. In the construction of G in the above reduction, we keep $H - V(R(H))$ as isolated satellite gadget $S(G)$ which is also added to X. With this modification, and the argument in the above reduction, the general case can be shown to be $\mathcal{NP}$-hard. □

Lemma 11 *If there exists (≤ 2)-all stars in $\mathcal{H}_d$, each of which contains exactly one S_2 and at least one S_1 as its connected components, and $\mathcal{H}$ is free from the following forbidden subgraphs, then* D-Π-VD *is $\mathcal{NP}$-hard:*

1. *A star $S_{\leq 2}$.*
2. *A (≤ 2)-all star H with exactly one occurrence of star S_2 and no S_1.*
3. *A (< 2)-all star.*
4. *A forbidden subgraph H which is an induced subgraph of $G' = (V', E')$, where V' can be partitioned into two sets M and J, such that:*
   i. *M induces a clique in G'.*
   ii. *For each $v \in J$, deg(v)=2 and $N(v) \subseteq M$.*
   iii. *Each vertex in M has exactly two neighbours in J.*
5. *A disconnected forbidden subgraph which consists of graphs in (1-4) above as its connected components.*

Proof: We first show how D-Π-VD is $\mathcal{NP}$-hard when one of the graphs in $\mathcal{H}_d$ is a disjoint union of a P_3 and a P_2, i.e. $P_3 + P_2$. Later, we show how this proof can be adapted for the general case. The proof is by reduction from 2-IN-3 SAT to D-Π-VD.

$\mathcal{NP}$-complete 2-IN-3 SAT [21] is a variant of 1-IN-3 SAT where now a truth assignment of Boolean values makes exactly two literals in each clause TRUE instead of one. Starting with an empty graph G and $X := \emptyset$, construct an instance (G, X) for D-Π-VD as follows. For each variable y_i, introduce a path Y_i of three vertices (variable gadget), add the middle vertex of Y_i to X and label the other vertices of Y_i with "+" and "−", respectively. For each clause c_j, add a star C_j with three leaves (clause gadget) and add its center vertex to X. Each of the three leaves of C_j corresponds to a literal in c_j, and each leaf is connected to a variable gadget as follows. Suppose that l_j^k is a literal y_j or $\neg y_j$, and let a_t be the leaf of C_j corresponding to l_j^k. Add a path M_t (connection gadget) on three vertices, identify one of its end vertices with a_t and another with vertex of Y_j labelled "+", if l_j^k is positive literal; otherwise, with the one labelled "−". Add all possible edges between the central vertices of connection gadgets. Finally, add an isolated path P of two vertices to G, and add it into X. An example of the following construction is given in Figure 7.

Obviously, $G[V \setminus X]$ only contains a clique M which consists of all the central vertices of connection gadgets M_t and two degree-2 vertices $\{x, y\}$ attached to each vertex of M such that the neighbourhood of all such vertices $\{x, y\}$ belongs in M (refer to Figure 7). Therefore, $G[V \setminus X] \in \Pi$. Now, with the above construction and arguments similar to the proof in Lemma 10, we can show that formula F has a satisfying truth assignment if and only if there exists a size-$(p + 3q)$ set X', $X' \cap X = \emptyset$, that obstructs every forbidden induced subgraphs in G. In other words, F has a satisfying truth assignment if and only if (G, X) is a YES-instance of D-Π-VD with solution size $p + 3q$. □

Finally, we present a polynomial-time solvable case of D-Π-VD with $\mathcal{H}$ containing (≤ 1)-all star; this case generalizes the result for DISJOINT SPLIT VERTEX DELETION, implicitly shown in [11]. We call a graph *pseudo-split* if the forbidden subgraph set $\mathcal{H}$ is equal to $\{2K_2, C_4\}$. Here C_4 is a cycle on four vertices.

Lemma 12 D-Π-VD *when* Π *is pseudo-split graphs can be solved in polynomial time.*

Proof:

A graph is pseudo-split if its vertex set can be partitioned into three possibly empty sets C, S and I [17], such that:

(a) C is a complete graph, I is an independent set and S (if non-empty) induces a C_5;

(b) Each vertex in C is global to S;

(c) There exists no edge between I and S.

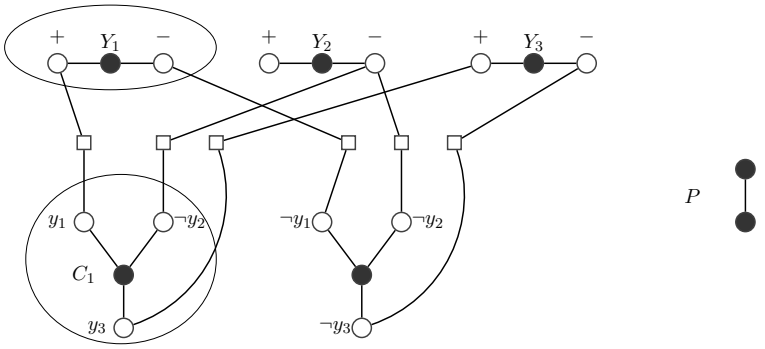

Figure 7: Example for the reduction in the proof of Lemma 11 from the 2-IN-3 SAT
instance with formula $(y_1 \wedge \neg y_2 \wedge y_3) \vee (\neg y_1 \wedge \neg y_2 \wedge \neg y_3)$ to DISJOINT Π-VERTEX DELETION with the given solution X (black vertices). The vertices in clique M is illustrated by square boxes. For illustration the the clause gadget C_1 and variable gadget Y_1 are labelled.

Such a partition is known as *pseudo-split partition*. We call a set $S \subseteq V$ a *pseudo-split vertex deletion set* if the graph $G[V \setminus S]$ is a pseudo-split graph.

In the following, we design a polynomial-time algorithm to solve D-Π-VD where $\mathcal{H} = \{2K_2, C_4\}$. Let the input instance be $I := (G = (V, E), X)$. If $G[X]$ contains an induced C_5, then $G[X]$ must have a unique partition. Let that unique partition be $(C_0 \biguplus I_0 \biguplus S_0)$. On the other hand, if $G[X]$ is C_5-free, then $G[X]$ is a split graph. In that case, $G[X]$ can have at most $x + 1$ split partitions $(C_0 \biguplus I_0)$ with $x = |X|$ [11]. Now, we guess and fix a partition $(C_0 \biguplus I_0 \biguplus S_0)$ where S_0 could be empty. The fixed partition should correspond to the pseudo-split partition of $G[V \setminus X']$. Hence, it remains to find the pseudo-split vertex deletion set of minimum size which is disjoint from X, and results in a pseudo-split graph with a partition which is consistent with $(C_0 \biguplus I_0 \biguplus S_0)$. More formally, we have an instance of the following problem:

Disjoint Pseudo-Split Vertex Deletion

Input: Graph $G = (V, E)$, a pseudo-split vertex deletion set $X \subset V$ such that $G[X]$ is a pseudo-split graph, a pseudo-split partition $(C_0 \biguplus I_0 \biguplus S_0)$ for the graph $G[X]$.

Output: Pseudo-split vertex deletion set X', disjoint from X such that $G[V \setminus X']$ has a pseudo-split partition consistent with $(C_0 \biguplus I_0 \biguplus S_0)$.

Assume X' is a solution, and let $(C' \biguplus I' \biguplus S')$ be a fixed partition for the graph $G[V \setminus X']$ consistent with the pseudo-split partition $(C_0 \biguplus I_0 \biguplus S_0)$. Let $(C_1 \biguplus I_1 \biguplus S_1)$ be the pseudo-split partition of the graph $G[V \setminus X]$. We distinguish the following cases:

Firstly we assume that there exists a C_5 in $G[X]$. Clearly, at most one vertex of C_1 and at most two vertices of S_1 can lie in I', and at most one vertex of I_1 and at most two vertices of S_1 can lie in C'. Hence, we initially guess these at most six vertices, denoted by $\{v_1, v_2, \cdots, v_6\}$. Let $v_1 = C_1 \cap I'$, $\{v_2, v_3\} = S_1 \cap I'$, $v_4 = I_1 \cap C'$ and $\{v_5, v_6\} = S_1 \cap C'$. We move $\{v_1, v_2, v_3\}$ to I_1 and $\{v_4, v_5, v_6\}$ to C_1. For the sake of convenience we refer to the modified sets C_1 and I_1 also as C_1 and I_1. Now, let $I^* = I_0 \cup I_1$ and $C^* = C_0 \cup C_1$. Since there already exists a C_5 in X, it is clear that any vertex in I^*, which is a neighbor of a vertex in $I_0 \cup \{v_1, v_2, v_3\}$ or S_0, need to be deleted, and any vertex in C^*, which is not global to $C_0 \cup \{v_4, v_5, v_6\}$ or S_0 needs to be deleted. Let X' be the set of such vertices that needs to be deleted. It is easy to see that it is sufficient to delete the set X' to obtain the required pseudo-split graph.

Next we assume that there exists no C_5 in X. For this case we make further case distinctions:

1. **There exists no C_5 in $G[V \setminus X']$.**
   In this case, we can run the compression routine from the iterative compression algorithm for Split Vertex Deletion in [11], since $G[X]$ and the resulting graph $G[V \setminus X']$ are split graphs.

2. **There exists a C_5 in $G[V \setminus X']$.**

   i. **The C_5 in $G[V \setminus X']$ is formed by vertices from $G[V \setminus X]$.** In this case, the C_5 in $G[V \setminus X']$ must be the same C_5 induced by vertices in S_1. Hence, S_1 must not be in X'. Moreover, C_0 must be global to S_1 and I_0 must be non-adjacent to S_1; otherwise, this case would not be possible. Now, since we cannot delete edges and S_1 will become S', no vertex of C_1 can lie in I', and no vertex of I_1 can lie in C'. It is clear that each vertex in I_1, which is a neighbor to a vertex in I_0, needs to be deleted, and each vertex in C_1, which is not global to C_0, needs to be deleted. Let X' be the set of such vertices that need to be deleted. It is easy to see that it is sufficient to delete the set X' to obtain the required pseudo-split graph.

   ii. **The C_5 in $G[V \setminus X']$ is formed from a combination of vertices from X and $G[V \setminus X]$.** In this case, the C_5 formed in $G[V \setminus X']$ must be formed by at most two vertices from C_1, at most four vertices from S_1, at most two vertices from I_1, at most two vertices from C_0, and at most two vertices from I_0. Moreover, since we cannot delete edges, at most one vertex of C_1 can lie in I', and at most one vertex of I_1 can lie in C'. We can observe that the number of different guesses will be polynomial in the size of the graph. Let S' be such guessed C_5. The required vertices from C_0, I_0, C_1, I_1 and S_1 will then be moved to S_0. Let $v_1 = C_1 \cap I'$ and $v_4 = I_1 \cap C'$. We move v_1 to I_1 and v_2 to C_1. For the sake of convenience, we refer to the modified sets C_1 and I_1 also as C_1 and I_1. Now, let $I^* = I_0 \cup I_1$ and $C^* = C_0 \cup C_1$. It is clear

that every vertex in I^*, which is a neighbor to a vertex in $I_0 \cup \{v_1\}$ or a vertex in S_0, needs to be deleted. Moreover, every vertex in C^*, which is not global to $C_0 \cup \{v_2\}$ or S_0, needs to be deleted. Let X' be the set of such vertices that need to be deleted. It is easy to see that it is sufficient to delete the set X' to obtain the required pseudo-split graph.

It is clear that the above cases cover all possibilities, and hence we solve Disjoint Pseudo-Split Vertex Deletion correctly. □

4 Open Problems

First, the computational complexity of D-Π-VD for the case, when (≤ 1)-all stars are present in $\mathcal{H}$, is partially resolved, for instance, for Π being threshold graphs. Here, the set $\mathcal{H}$ consists of $2K_2$, C_4 and P_4. It would also be interesting to study the computational complexity of D-Π-VD in directed graphs and the variant of D-Π-VD where the solution comprises of edges to be deleted instead of vertices.

References

[1] V. Bafna, P. Berman, and T. Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discrete Math.*, 12(3):289–297, 1999. doi:10.1137/S0895480196305124.

[2] J. Chen, F. V. Fomin, Y. Liu, S. Lu, and Y. Villanger. Improved algorithms for the feedback vertex set problems. In *WADS*, pages 422–433, 2007. doi:10.1007/978-3-540-73951-7_37.

[3] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *STOC*, pages 177–186, 2008. doi:10.1145/1374376.1374404.

[4] F. K. H. A. Dehne, M. R. Fellows, M. A. Langston, F. A. Rosamond, and K. Stevens. An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. *Theory Comput. Syst.*, 41(3):479–492, 2007. doi:10.1007/s00224-007-1345-z.

[5] R. Diestel. *Graph theory*. Springer-Verlag, New York, 2 edition, 2000.

[6] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, NY, 1999. 530 pp.

[7] M. R. Fellows, J. Guo, H. Moser, and R. Niedermeier. A complexity dichotomy for finding disjoint solutions of vertex deletion problems. *TOCT*, 2(2):5, 2011. doi:10.1145/1944857.1944860.

[8] H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for network problems. *SIAM J. Comput.*, 18:1013–1036, 1989. doi:10.1137/0218069.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[10] S. Gaspers and M. Mnich. Feedback vertex sets in tournaments. *Journal of Graph Theory*, 72(1):72–89, 2013. doi:10.1002/jgt.21631.

[11] E. Ghosh, S. Kolay, M. Kumar, P. Misra, F. Panolan, A. Rai, and M. S. Ramanujan. Faster parameterized algorithms for deletion to split graphs. In *SWAT*, pages 107–118, 2012. doi:10.1007/978-3-642-31155-0_10.

[12] J. Guo. Problem kernels for NP-complete edge deletion problems: Split and related graphs. In *ISAAC*, pages 915–926, 2007. doi:10.1007/978-3-540-77120-3_79.

[13] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.*, 72(8):1386–1396, 2006. doi:10.1016/j.jcss.2006.02.001.

[14] J. Guo, H. Moser, and R. Niedermeier. Iterative compression for exactly solving NP-hard minimization problems. In *Algorithmics of Large and Complex Networks*, pages 65–80, 2009. doi:10.1007/978-3-642-02094-0_4.

[15] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. In *LATIN*, pages 711–722, 2008. doi:10.1007/978-3-540-78773-0_61.

[16] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

[17] F. Maffray and M. Preissmann. Linear recognition of pseudo-split graphs. *Discrete Applied Mathematics*, 52(3):307–312, 1994. doi:10.1016/0166-218X(94)00022-0.

[18] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109–128, 2001. doi:10.1016/S0166-218X(00)00391-7.

[19] R. Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, March 2006.

[20] B. A. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

[21] T. J. Schaefer. The complexity of satisfiability problems. In *STOC*, pages 216–226, 1978. doi:10.1145/800133.804350.