A Complexity Dichotomy for Finding Disjoint Solutions of Vertex Deletion Problems

Michael R. Fellows[1,⋆], Jiong Guo[2,⋆⋆], Hannes Moser[2,⋆⋆⋆], and Rolf Niedermeier[2]

[1] PC Research Unit, Office of DVC (Research), University of Newcastle, Callaghan, NSW 2308, Australia
michael.fellows@newcastle.edu.au

[2] Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{jiong.guo,hannes.moser,rolf.niedermeier}@uni-jena.de

Abstract. We investigate the computational complexity of a general "compression task" centrally occurring in the recently developed technique of iterative compression for exactly solving NP-hard minimization problems. The core issue (particularly but not only motivated by iterative compression) is to determine the computational complexity of, given an already inclusion-minimal solution for an underlying (typically NP-hard) vertex deletion problem in graphs, to find a better *disjoint* solution. The complexity of this task is so far lacking a systematic study. We consider a large class of vertex deletion problems on undirected graphs and show that, except for few cases which are polynomial-time solvable, the others are NP-complete. This class includes problems such as VERTEX COVER (here the corresponding compression task is decidable in polynomial time) or UNDIRECTED FEEDBACK VERTEX SET (here the corresponding compression task is NP-complete).

1 Introduction

With the introduction of the iterative compression by Reed et al. [17] in 2004, parameterized complexity analysis has gained a new tool for showing fixed-parameter tractability results for NP-hard minimization problems (cf. [9, 15]). For instance, in 2008, applying iterative compression has led to major breakthroughs concerning the classification of the parameterized complexity of two important problems. First, Chen et al. [4] showed that the NP-complete DIRECTED FEEDBACK VERTEX SET problem is fixed-parameter tractable. Second, Razgon and O'Sullivan [16] proved that the NP-complete ALMOST 2-SAT

⋆ Supported by the Australian Research Council. Work done while staying in Jena as a recipient of the Humboldt Research Award of the Alexander von Humboldt Foundation, Bonn, Germany.
⋆⋆ Partially supported by the DFG, PALG, NI 369/8.
⋆⋆⋆ Supported by the DFG, project AREG, NI 369/9.

problem is fixed-parameter tractable. Refer to the recent survey [9] for more on iterative compression applied to exactly solving NP-hard minimization problems.

The central idea behind iterative compression is to employ a *compression routine*. This is an algorithm that, given a problem instance and a corresponding solution, either calculates a smaller solution or proves that the given solution is of minimum size. Using a compression routine, one finds an optimal solution to a problem by inductively building up the problem instance and iteratively compressing intermediate solutions. Herein, the essential fact from the viewpoint of parameterized complexity is that if the task performed by the compression routine is fixed-parameter tractable, then so is the problem solved by means of iterative compression. The main strength of iterative compression is that it allows to see the problem from a different angle: The compression routine does not only have the problem instance as input, but also a solution, which carries valuable structural information. The design of a compression routine, therefore, may be simpler than showing that the original problem is fixed-parameter tractable.

While embedding the compression routine into the iteration framework is usually straightforward, finding the compression routine itself is not [9, 15]. For many vertex deletion problems, a common approach to designing a compression routine is to branch on the possible subsets of the uncompressed solution to retain in the compressed solution. This leads to the following generic problem that asks for a *disjoint* compressed solution:

Input: An instance of the underlying NP-hard problem and a solution S.[3]
Question: Is there a solution S' such that $S' \cap S = \emptyset$ and $|S'| < |S|$?

We study the complexity of COMPRESSION TASK depending on what the underlying NP-hard problem is. The computational complexity of COMPRESSION TASK, so far, remains widely unclassified. For instance, the fixed-parameter tractability results (using iterative compression) for VERTEX BIPARTIZATION [17] or UNDIRECTED FEEDBACK VERTEX SET [3, 5, 8] leave open whether the respective COMPRESSION TASK is NP-hard or polynomial-time solvable. By way of contrast, the fixed-parameter tractability result for the NP-complete CLUSTER VERTEX DELETION problem [10] is based on a polynomial-time algorithm for COMPRESSION TASK. Here, extending a framework attributed to Yannakakis [12], we contribute a complete classification of COMPRESSION TASK for a natural class of vertex deletion problems (specified by a graph property Π), including all of the above mentioned problems.

A *graph property* Π is a set of graphs; in the following, we say that a graph G *satisfies* Π if $G \in \Pi$. 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.

The classical Π-VERTEX DELETION problem is defined as follows: for a nontrivial hereditary graph property Π testable in polynomial time, given an undirected graph G and a positive integer k, decide whether it is possible to delete at most k vertices from the graph such that the resulting graph satisfies Π.

[3] Here, the solution is a set. It is conceivable that the COMPRESSION TASK can be formulated also for other types of solutions.

For example, Undirected Feedback Vertex Set corresponds to the case that Π means "being cycle-free". Yannakakis has shown that Π-Vertex Deletion is NP-complete for any non-trivial hereditary graph property Π in general graphs [12]. General vertex deletion problems have also been studied in terms of their parameterized complexity [2, 11].

The Compression Task restricted to vertex deletion problems with property Π, called Disjoint Π-Vertex Deletion, can be formulated as follows:

Input: An undirected graph $G = (V, E)$ and a vertex subset $S \subseteq V$ such that $G[V \setminus S]$ satisfies Π and S is inclusion-minimal under this property, that is, for every proper subset $S' \subset S$ the graph $G[V \setminus S']$ does not satisfy Π.
Question: Is there a vertex subset $S' \subseteq V$ of size at most $|S|$, such that $S \cap S' = \emptyset$ and $G[V \setminus S']$ satisfies Π?

We replace the requirement $|S'| < |S|$ in the definition of Compression Task by $|S'| \leq |S|$ without changing the computational complexity, because the corresponding hardness reductions (cf. Lemma 4) work for both cases, and the last case might be of interest if S is already optimal. Moreover, we demand that S is inclusion-minimal; any solution can be made inclusion-minimal in polynomial time if Π can be tested in polynomial time. Thus, this requirement does not change the complexity.

A graph property Π is *determined by the components* if it holds that if every connected component of the graph satisfies Π, then so does the whole graph. The central result of this work can be informally stated as follows:

Main Theorem: *Let Π be any non-trivial hereditary graph property that is determined by the components and that can be tested in polynomial time.* Disjoint Π-Vertex Deletion *is NP-complete unless Π is the set of all graphs whose connected components are cliques or Π is the set of all graphs whose connected components are cliques of at most s vertices, $s \geq 1$—in these cases it is polynomial-time solvable.*[4]

The main theorem applies to many natural vertex deletion problems in undirected graphs, including Vertex Cover, Bounded-Degree Deletion, Undirected Feedback Vertex Set [3, 5, 8], Vertex Bipartization [17], Cluster Vertex Deletion [10], Chordal Deletion [13], and Planar Deletion [14]. Thus, except for Vertex Cover and Cluster Vertex Deletion, all other problems have an NP-complete Compression Task.

Our original motivation for this work comes from the desire to better understand the limitations of the iterative compression technique. Beyond this, Disjoint Π-Vertex Deletion also seems to be a natural and interesting problem on its own: In combinatorial optimization, one often may be confronted with finding *alternative* good solutions to already found ones. In the setting of Disjoint Π-Vertex Deletion, this is put to the extreme in the sense that we ask for solutions that are completely unrelated, that is, disjoint. For instance, this demand also naturally occurs in the context of finding quasicliques [1]. Due to the lack of space, some proofs are deferred to a full version of the paper.

[4] There might exist other polynomial-time solvable cases for *non-hereditary* properties.

Preliminaries. We only consider undirected graphs $G = (V, E)$ with $n := |V|$ and $m := |E|$. We write $V(G)$ and $E(G)$ to denote, respectively, the vertex and edge set of a graph G. For $v \in V$, let $N_G(v) := \{u \in V \mid \{u, v\} \in E\}$ and let $\deg_G(v) := |N_G(v)|$. For $S \subseteq V$, let $N_G(S) := \bigcup_{v \in S} N(v) \setminus S$. For $S \subseteq V$, let $G[S]$ be the subgraph of G induced by S and $G - S := G[V \setminus S]$. For $v \in V$, let $G - v := G[V \setminus \{v\}]$. For a connected graph G, a *cut-vertex* is a vertex $v \in V$ such that $G - v$ is not connected. A K_3 is a complete graph on three vertices. For $s \geq 1$, the graph $K_{1,s} = (\{u, v_1, \ldots, v_s\}, \{\{u, v_1\}, \ldots, \{u, v_s\}\})$ is a *star*. The vertex u is the *center* of the star and the vertices $v_1, \ldots, v_s$ are the *leaves* of the star.

If a graph H does not satisfy some hereditary property Π, then any supergraph of H does not satisfy Π. We call H a *forbidden subgraph* for Π. For any hereditary property Π there exists a set $\mathcal{H}$ of "minimal" forbidden induced subgraphs, that is, forbidden graphs for which every induced subgraph satisfies Π [7]. For this work, we restrict our attention to non-trivial hereditary properties that are determined by the components. For the corresponding characterization of Π by forbidden induced subgraphs, this means that the set of forbidden subgraphs only contains *connected* graphs.

By simple counting arguments, there exist Disjoint Π-Vertex Deletion problems that are not in NP. As Lewis and Yannakakis [12], we add the stipulation that Π can be tested in polynomial time, hence the corresponding Disjoint Π-Vertex Deletion problem is in NP, and our hardness results to come thus will show that it is NP-complete.

A parameterized problem (I, k) is *fixed-parameter tractable* with respect to the parameter k if it can be solved in $f(k) \cdot \text{poly}(|I|)$ time, where I is the input instance and f is some computable function. The corresponding algorithm is called *fixed-parameter algorithm*.

2 Polynomial-Time Solvable Cases

This section covers all cases of Disjoint Π-Vertex Deletion that can be decided in polynomial time. The corresponding graph properties are as follows:

Definition 1. *Let Π_s, for $s \geq 1$, be the graph property that contains all graphs whose connected components are cliques of at most s vertices. Furthermore, let Π_∞ be the graph property that contains all graphs whose connected components of G are cliques (of arbitrary size).*

For instance, Π_1, Π_2, and Π_∞ are the properties "being edgeless", "being a graph of maximum degree one", and "every connected component is a clique (of arbitrary size)", respectively. The corresponding sets of forbidden induced subgraphs consist of a single edge (Π_1), a path on three vertices and a clique on three vertices (Π_2), and a path on three vertices (Π_∞). In general, the set of forbidden induced subgraphs of Π_s for $s \geq 2$ contains a path on three vertices and an $(s+1)$-vertex clique. Summarizing, for each property Π_s, $s \geq 1$, and Π_∞, the corresponding set of forbidden induced subgraphs contains a star with at most two leaves, and these are clearly the only properties whose sets of forbidden induced subgraphs contain a star with at most two leaves.

Theorem 1. Disjoint Π-Vertex Deletion *is decidable in polynomial time if $\Pi = \Pi_s$, for some $s \geq 1$, or if $\Pi = \Pi_\infty$.*

Concerning property Π_1, obviously, there can only exist a disjoint solution S', $S' \cap S = \emptyset$, if S forms an independent set in G. Moreover, S' must contain every endpoint of each edge that has one endpoint in S and the other endpoint in $V \setminus S$. Hence, the input is a yes-instance iff S forms an independent set and $|N_G(S)| \leq |S|$. This condition can be tested in polynomial time.

Lemma 1. Disjoint Π_1-Vertex Deletion *can be decided in polynomial time.*

Disjoint Π_∞-Vertex Deletion is equivalent to the decision version of the compression step for Cluster Vertex Deletion [10].

Lemma 2 ([10]). Disjoint Π_∞-Vertex Deletion *can be decided in polynomial time.*

The polynomial-time decidability for the remaining properties Π_s can be proven with similar techniques as in the proof of Lemma 2.

Lemma 3. *For each $s \geq 2$,* Disjoint Π_s-Vertex Deletion *can be decided in polynomial time.*

3 NP-Hardness Framework and Simple Proofs

Lewis and Yannakakis [12] showed that Π-Vertex Deletion for any non-trivial hereditary property Π is NP-complete. Due to the similarity of Π-Vertex Deletion to Disjoint Π-Vertex Deletion, in some simple cases we can adapt the framework from [12].[5] This section is mainly devoted to this framework and how it is modified to partially use it for Disjoint Π-Vertex Deletion.

There are cases, however, where adapting this framework fails; this happens when there is a star with at least three leaves among the family $\mathcal{H}$ of forbidden induced subgraphs, because (as we will see later) a star with at least three leaves does not permit to derive a given solution S for the graph that is constructed by the reduction of the framework. For this case, we have to devise other NP-hardness proofs (if there is a star with at most two leaves, then the problem is polynomial-time decidable). Summarizing, we have to distinguish the following three cases (recall that each graph in $\mathcal{H}$ is connected): (1) $\mathcal{H}$ does not contain a star (NP-hard, this section, Theorem 2), (2) $\mathcal{H}$ contains a star with at least three leaves (NP-hard, Section 4, Theorem 3), and (3) $\mathcal{H}$ contains a star with at most two leaves (polynomial-time decidable, Section 2, Theorem 1).

The main result of this section covers all cases that can be proven by adapting the framework by Yannakakis as described in the remainder of this section.

[5] As made explicit in Lewis and Yannakakis' paper [12], the parts of it we are referring to in our work have been contributed by Yannakakis.

Theorem 2. *Let Π be a non-trivial hereditary property that is determined by the components and let $\mathcal{H}$ be the corresponding set of all forbidden induced subgraphs. If $\mathcal{H}$ contains no star, then* DISJOINT Π-VERTEX DELETION *is NP-hard.*

The Framework by Yannakakis and its Limitations. In the following, we briefly describe the reduction by Yannakakis [12], which shows that any vertex deletion problem with non-trivial hereditary graph property is NP-hard. Since the hereditary graph properties considered in this paper are assumed to be determined by the components, we present a variant that is restricted to such properties, that is, the forbidden induced subgraphs shall be connected.

Preliminaries. Let $\mathcal{H}$ be the set of forbidden induced subgraphs that correspond to the non-trivial hereditary property Π that is determined by its components. An important concept for the framework is the notion of α-sequences [12].

Definition 2 (α-sequence). *For a connected graph $H \in \mathcal{H}$, if H is 1-connected, then take a cut-vertex c and sort the components of $H - c$ according to their size. If H is not 1-connected, then let c be an arbitrary vertex (in this case, $H - c$ has just one connected component). Sorting the connected components of $H - c$ with respect to their sizes gives a sequence $\alpha = (n_1, \ldots, n_i)$, where $n_1 \geq \ldots \geq n_i$. The sequence depends on the choice of c. The* α-sequence *of H, $\alpha(H)$, is a sequence which yields a lexicographically smallest such sequence α.*

Let $H \in \mathcal{H}$ be a graph with lexicographically smallest α-sequence among all graphs in $\mathcal{H}$. Note that every induced subgraph of H has a lexicographically smaller α-sequence than H. Since Π is satisfied by all independent sets, the connected graph H must contain at least two vertices, thus a largest component J of $H - c$ contains at least one vertex. Let d be an arbitrary vertex in J, and let H' be the graph resulting by removing all vertices in J from H, and let J' be the graph induced by $V(J) \cup \{c\}$ in H.

Reduction. The reduction by Yannakakis [12] from the NP-complete VERTEX COVER[6] problem works as follows. Let G be an instance of VERTEX COVER. For every vertex v in G create a copy of H' and identify c and v. Replace every edge $\{u, v\}$ in G by a copy of J', identifying c with u and d with v. Let G' be the resulting graph.

Correctness. The graph G has a size-k vertex cover if and only if G' has a size-k vertex set that obstructs all forbidden induced subgraphs $\mathcal{H}$ in G':

($\Rightarrow$) If A is a vertex cover of G, then $S' := A$ also obstructs all graphs in $\mathcal{H}$: Every connected component of $G' - S'$ is either (1) a copy of $H' - c$ or (2) a copy of H' together with several copies of J', each with either c or d deleted. In the latter case, the copy of H' and the copies of J' intersect exactly in one vertex of $V(G)$. Let C be such a connected component and let v be the described vertex. In case (1), $\alpha(H' - c)$ is lexicographically smaller than $\alpha(H)$

[6] Given a graph $G = (V, E)$ and $k \geq 0$, decide whether there exists a set $S \subseteq V$ of size at most k such that each edge has at least one endpoint in S.

since $H'-c$ is a subgraph of H. In case (2), v is a cut-vertex and the components of $C-v$ can be divided into a copy of $H'-c$ and several copies of J with one vertex deleted. Since the latter type of components has less than $|V(J)|$ vertices, the cut-vertex v gives an α-sequence for C which is lexicographically smaller than the α-sequence of H. Thus, the connected components in $G'-S'$ have a smaller α-sequence than H, and because H is a forbidden induced subgraph with lexicographically smallest α-sequence, these connected components do not contain forbidden induced subgraphs.

($\Leftarrow$) If S' is a solution for $\mathcal{H}$-DELETION, then one can determine a vertex cover A for G: for each $w \in S'$, if w is in a copy of H' (possibly $w \in V(G)$), then add vertex c of that copy of H' to A, and if w is in a copy of J' (where $w \notin V(G)$), then add vertex c of that copy of J' to A. Obviously, $|A| \leq |S'|$. Suppose that there exists an edge $\{u, v\}$ in $G-A$. Then, S' neither contains any vertex from the two copies of H' corresponding to the vertices u and v nor from the copy of J' that replaced the edge $\{u, v\}$ in the construction of G'. Hence $G'-A$ contains a copy of H, a contradiction. Therefore, A is a vertex cover for G.

Limitations. In some cases, a very similar reduction principle can be applied for DISJOINT Π-VERTEX DELETION. We simply have to show that there exists an $\mathcal{H}$-obstruction set S in G' with the only restriction that S does not contain any vertex from $V(G)$. Then, in principle, we can use the same arguments as above. However, for some cases this approach fails; for instance, if J' is a clique and some graph of $\mathcal{H}$ is contained in G, then this forbidden induced subgraph, which also exists in G', can only be obstructed by vertices in $V(G)$. For example, this happens when Π is the property "being cycle-free" (FEEDBACK VERTEX SET): $\mathcal{H}$ contains all cycles, and the graph H with the smallest α-sequence is K_3. One can deal with this situation by reducing from K_3-free graphs, and using the graph with the smallest α-sequence among all K_3-free graphs in H, as shown in the proof of Lemma 5. The same type of problem, however, also occurs if H is a star. In this case, each connected component of $H-c$ is an isolated vertex. Thus, the vertex d has to be one of these vertices, and G and therefore G' might contain a forbidden induced subgraph with lexicographically higher α-sequence than H. This induced subgraph cannot be obstructed by a set S that is not allowed to contain any vertex from $V(G)$. In this case, the framework by Yannakakis cannot be used and we have to devise other reduction techniques (Section 4).

New Proofs Based on the Reduction Framework by Yannakakis. Recall that we assume here that the set of forbidden induced subgraph corresponding to Π contains no star. We have to distinguish between the cases that (1) all forbidden induced subgraphs contain a K_3 (see Lemma 4), and that (2) not all forbidden induced subgraph contain a K_3 (see Lemma 5).

Lemma 4. *If the set $\mathcal{H}$ of forbidden induced subgraphs corresponding to Π contains only graphs that contain a K_3, then* DISJOINT Π-VERTEX DELETION *is NP-hard.*

Proof. The proof is by reduction from the NP-complete VERTEX COVER on K_3-free graphs [6]. Let (G, k) be an instance of VERTEX COVER, where G is K_3-free.

First, construct a graph G' using the reduction scheme by Yannakakis. Greedily compute a minimal $\mathcal{H}$-obstruction set S_1 for G' such that $S_1 \cap V(G) = \emptyset$. Such a set S_1 always exists, since G is K_3-free and, therefore, does not contain any forbidden induced subgraph.

It remains to take care of the size of the new solution S'; recall that Disjoint Π-Vertex Deletion asks for a solution S' such that $|S'| \leq |S|$. First, suppose that $k \leq |S_1|$. Informally speaking, we have to force that only k vertices out of the $|S_1|$ available vertices can be used in G' to obstruct all forbidden induced subgraphs. Let H, c, J, J', and d be defined as in the reduction scheme. We add a padding gadget C constructed as follows to G'. Add a new vertex w and $|S_1| - k + 1$ copies of H to G', identify the vertex d of each newly added copy of H with w, and let $S := S_1 \cup \{w\}$. The gadget C is obviously connected and w is a cut-vertex in C. The vertex w obstructs all forbidden induced subgraphs in C, because deleting w (and, thus, d) from each copy of H in C leaves a graph with lexicographically smaller α-sequence (witnessed by c in each copy of H). Hence, S is a minimal $\mathcal{H}$-obstruction set for G'.

An $\mathcal{H}$-obstruction set S' for G' with $S' \cap S = \emptyset$ must contain at least one vertex in each copy of H in C, thus S' must contain at least $|S_1| - k + 1$ vertices of C; putting into S' the vertex c of each copy of H in C obstructs every forbidden induced subgraph in H: every connected component of $C - S'$ either is a copy of $H - c$ or consists of $|S_1| - k + 1$ copies of J that pairwise overlap in vertex w. In the latter case, w is a cut-vertex witnessing that each remaining connected component has size smaller than J, yielding a lexicographically smaller α-sequence. This shows that S', in order to obstruct all forbidden induced subgraphs in C, contains at least $|S_1| - k + 1$ vertices. Since $S = S_1 \cup \{w\}$, there remain at most $|S| - |S_1| + k - 1 = k$ vertices to obstruct all forbidden induced subgraphs in $G' - V(C)$.

If $|S_1| < k$, then construct C in the same manner with $k - |S_1| + 1$ copies of H and let S be the union of S_1 and the vertex c of each copy of H. Then, the new solution S' can obstruct all forbidden induced subgraphs in C with the vertex w, and there are $k - |S_1| + 1 + |S_1| - 1 = k$ vertices left to obstruct all forbidden induced subgraphs in $G' - V(C)$.

By these arguments and the reduction scheme, G has a size-k vertex cover if and only if G' has a $\mathcal{H}$-obstruction set S' with $S' \cap S = \emptyset$ and $|S'| \leq |S|$. $\square$

In the following, assume that not all forbidden induced subgraphs contain a K_3, let $\mathcal{H}' \subseteq \mathcal{H}$ be the set of all forbidden induced subgraphs that do not contain a K_3, and let H be a forbidden induced subgraph with lexicographically smallest α-sequence among all graphs in $\mathcal{H}'$.

Lemma 5. *If the set $\mathcal{H}$ of forbidden induced subgraphs corresponding to Π contains no stars, but other graphs that do not contain a K_3, then* Disjoint Π-Vertex Deletion *is NP-hard.*

4 Refined Reduction Strategies

Here, we present NP-hardness proofs if H is a star with at least three leaves. The main result of this section is as follows.

Theorem 3. *Let Π be a non-trivial hereditary graph property that is determined by the components and let $\mathcal{H}$ be the corresponding set of all forbidden induced subgraphs. If $\mathcal{H}$ contains a star with at least three leaves, then* DISJOINT Π-VERTEX DELETION *is NP-hard.*

Note that a star has a smaller α-sequence than any other forbidden induced subgraph that is not a star, and there is only one star in $\mathcal{H}$, since the graphs in $\mathcal{H}$ are inclusion-minimal. Therefore, if $\mathcal{H}$ contains a star, then the graph with smallest α-sequence is necessarily the star in $\mathcal{H}$. Let H be the star in $\mathcal{H}$.

The proof of Theorem 3 is based on the following case distinction. (1) H is a star with at least four leaves (Lemma 6), or (2) H is a star with three leaves. In the latter case, we distinguish the following two subcases: (2a) $\mathcal{H}$ contains a P_4 (Lemma 7), and (2b) $\mathcal{H}$ does not contain a P_4 (Lemma 8).

Lemma 6. *If the set $\mathcal{H}$ of forbidden induced subgraphs corresponding to property Π contains a star H, and if H has at least four leaves, then* DISJOINT Π-VERTEX DELETION *is NP-hard.*

Next, we show the NP-hardness of the case that the smallest graph in the set of forbidden induced subgraphs is a star with three leaves. In this case, a reduction from VERTEX COVER seems less promising, since the VERTEX COVER instance we reduce from contains vertices of degree three and therefore copies of the forbidden induced star with three leaves. Hence, we use 3-CNF-SAT. First, we consider the case that the path on four vertices is also forbidden.

Lemma 7. *If the set $\mathcal{H}$ of forbidden induced subgraphs corresponding to property Π contains a star H, and if H has three leaves and $\mathcal{H}$ also contains the path on four vertices, then* DISJOINT Π-VERTEX DELETION *is NP-hard.*

Proof. The proof is by reduction from 3-CNF-SAT. Let $F = c_1 \wedge \cdots \wedge c_m$ be a 3-CNF formula over a variable set $X = \{x_1, \ldots, x_n\}$. We denote the kth literal in clause c_j by l_j^k, for $1 \leq k \leq 3$. An example of the following construction is given in Figure 1. Starting with an empty graph G and $S := \emptyset$, construct an instance (G, S) for DISJOINT Π-VERTEX DELETION as follows. For each variable x_i, introduce a cycle X_i of $12m$ vertices (variable gadget), add every second vertex on X_i to S, and label all the other vertices on the cycle alternately with "+" and "−". For each clause c_j, add a star C_j with three leaves (clause gadget) and add its center vertex to S. 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 x_i or $\neg x_i$, and let a_k be the leaf of C_j corresponding to l_j^k. Add a star with three leaves (connection gadget), identify one leaf with a_i, identify another leaf with an *unused vertex*[7] on X_i with label "+" if l_j^k is positive

[7] This means that no vertex of an other connection gadget has been identified with this vertex on X_i, that is, it is of degree two.

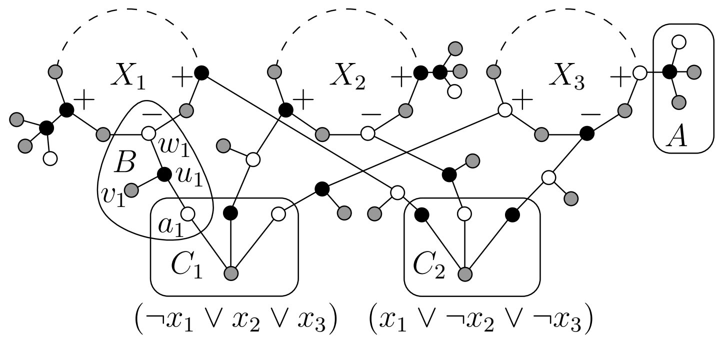

Fig. 1: Example for the reduction in the proof of Lemma 7 for the 3-CNF-SAT formula $(\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3)$. For illustration, one minimality gadget is labeled with A and one connection gadget is labeled with B. Furthermore, for the connection gadget B the vertices are named according to the definitions of a_k, u_k, v_k and w_k in the proof of Lemma 7 for $k = 1$. The vertices in the given solution S are gray, the vertices in the disjoint solution S', corresponding to the satisfying truth assignment $x_1 =$ true, $x_2 =$ true, $x_3 =$ false, are black.

and with an unused vertex on X_i with label "−" if l_j^k is negative, and add the remaining leaf to S. Finally, for each remaining unused vertex v labeled "+" or "−" in G, add a star with three leaves (minimality gadget), add two of its leaves to S, and add an edge between the center and v. This concludes the construction.

Obviously, $G - S$ only contains paths on three vertices as connected components (cf. Figure 1), that is, $G - S$ satisfies Π. Moreover, S is minimal, that is, for any $v \in S$, $G - (S \setminus \{v\})$ does not satisfy Π. Let q be the number of minimality gadgets. We show that formula F has a satisfying truth assignment if and only if there exists a size-$(q+3nm+3m)$ set S', $S' \cap S = \emptyset$, that obstructs all forbidden induced subgraphs in G. Analogously to the proof of Lemma 6, the construction can be modified (to "correct" the sizes of S and S') by adding a padding gadget based on stars with three leaves. This straightforward modification is omitted.

($\Rightarrow$) We defer the proof of this direction to a full version of the paper.

($\Leftarrow$) Let S', $S' \cap S = \emptyset$, be a size-$(q+3nm+3m)$ vertex set that obstructs every forbidden induced subgraph in G. We may assume that S' does not contain any degree-one vertex in G (since a degree-one vertex in S' could be simply replaced by its neighbor). Recall that the set of forbidden induced subgraphs contains the star with three leaves and the path on four vertices. Each minimality gadget is a star with three leaves, and since we assumed that no degree-one vertex is in S', its center vertex must be in S'. Hence, S' contains exactly q vertices of the minimality gadgets. Since P_4s are forbidden, at least every fourth vertex on the cycle of each variable gadget has to be in S'. However, we will see that S' contains exactly three vertices for each clause (thus, $3m$ vertices for all clauses), and these vertices cannot be vertices on any variable gadget. Therefore, for each variable gadget X_i, the set S' must contain *exactly* every fourth vertex of X_i (in order to obtain a total number of $3mn$ vertices in S' for all n variable gadgets), thus S' either contains all vertices labeled "+" or all vertices labeled "−". If S' contains all vertices labeled "+", then we set $x_i :=$ true, if S' contains all vertices labeled "−", then we set $x_i :=$ false. It remains to show that the assignment defined in this way is a satisfying truth assignment for the formula F.

For a clause gadget C_j, and for each leaf a_k of C_j corresponding to literal l_j^k, let u_k be the center of the corresponding connection gadget, v_k be the degree-one neighbor of u_k, and w_k be the neighbor of u_k on the variable gadget X_i, for some $1 \leq i \leq n$ (cf. Figure 1). There is a P_4 containing the center of C_j, together with a_k, u_k, and v_k. Since the center of C_j is in S, the set S' has to contain at least three vertices to obstruct the three P_4s corresponding to C_j (one for each leaf). Thus, for all clauses, there are at least $3m$ vertices in S' that obstruct these P_4s. In total, S' contains $q + 3nm + 3m$ vertices. Therefore, there are *exactly* $3m$ vertices in S' that obstruct these P_4s. Thus, for a clause gadget C_j, for each leaf a_k, either $a_k \in S'$ or $u_k \in S'$. Which case applies depends on which vertices from X_i are in S': if $w_k \notin S'$, then w_k together with u_k and its two neighbors on X_i induce a star with three leaves, thus $u_k \in S'$. If $w_k \in S'$, then either $a_k \in S'$ or $u_k \in S'$. If $w_k \in S'$ and $u_k \in S'$, however, then one can simply remove u_k from S' and add a_k instead. After that, S' still obstructs all forbidden induced subgraphs. Since S' obstructs all forbidden induced subgraphs, at least one leaf a_k of C_j must be in S', which implies that $w_k \in S'$. Let X_i be the variable gadget that contains w_k. If w_k has label "+", then $x_i =$ true by the definition of the assignment, and by construction $l_j^k = x_i$ is a positive literal, hence c_j is satisfied. If w_k has label "−", then $x_i =$ false, and, by construction, $l_j^k = \neg x_i$ is a negative literal, hence c_j is satisfied. Summarizing, for every clause there is at least one true literal and thus the constructed truth assignment satisfies F. $\square$

Finally, we consider the case that the path on four vertices is not forbidden.

Lemma 8. *If the set $\mathcal{H}$ of forbidden induced subgraphs corresponding to property Π contains a star H, and if H has three leaves and $\mathcal{H}$ does not contain the path on four vertices, then* DISJOINT Π-VERTEX DELETION *is NP-hard.*

5 Outlook

As indicated in the introductory section, there are important problems amenable to iterative compression that do not fall into the problem class studied here. Among these, in particular, we have DIRECTED FEEDBACK VERTEX SET and ALMOST 2-SAT. Hence, it would be interesting to further generalize our results to other problem classes, among these also being vertex deletion problems on directed graphs or bipartite graphs and edge deletion problems. Our work here has left open the case that a forbidden subgraph may consist of more than one connected component. Finally, one could explore to parameterize DISJOINT Π-VERTEX DELETION by the number of vertices by which S' should at least differ from S.

References

[1] J. Abello, M. G. C. Resende, and S. Sudarsky. Massive quasi-clique detection. In *Proc. 5th LATIN*, volume 2286 of *LNCS*, pages 598–612. Springer, 2002.

[2] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.

[3] J. Chen, F. V. Fomin, Y. Liu, S. Lu, and Y. Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. System Sci.*, 74(7):1188–1198, 2008.

[4] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008. Article 21, 19 pages.

[5] 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.

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

[7] D. L. Greenwell, R. L. Hemminger, and J. B. Klerlein. Forbidden subgraphs. In *Proc. 4th CGTC*, pages 389–394, 1973.

[8] 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. System Sci.*, 72(8):1386–1396, 2006.

[9] J. Guo, H. Moser, and R. Niedermeier. Iterative compression for exactly solving NP-hard minimization problems. In *Algorithmics of Large and Complex Networks*, volume 5515 of *LNCS*, pages 65–80. Springer, 2009.

[10] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory Comput. Syst.*, 2009. Available electronically.

[11] S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theor. Comput. Sci.*, 289(2):997–1008, 2002.

[12] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. System Sci.*, 20(2):219–230, 1980.

[13] D. Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 2009. Available electronically.

[14] D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. In *Proc. 33rd WG*, volume 4769 of *LNCS*, pages 292–303. Springer, 2007.

[15] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[16] I. Razgon and B. O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. System Sci.*, 2009. Available electronically.

[17] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.