Parameterized Complexity of Feedback Vertex Sets on Hypergraphs

Pratibha Choudhary
Indian Institute of Technology Jodhpur, Jodhpur, India, pratibhac247@gmail.com

Lawqueen Kanesh
Institute of Mathematical Sciences, HBNI, Chennai, India, lawqueen@imsc.res.in

Daniel Lokshtanov
University of California Santa Barbara, Santa Barbara, USA, daniello@ucsb.edu

Fahad Panolan
Indian Institute of Technology Hyderabad, India, fahad@cse.iith.ac.in

Saket Saurabh
Institute of Mathematical Sciences, HBNI, Chennai, India, saket@imsc.res.in

Abstract

A *feedback vertex set* in a hypergraph H is a set of vertices S such that deleting S from H results in an acyclic hypergraph. Here, deleting a vertex means removing the vertex and all incident hyperedges, and a hypergraph is *acyclic* if its vertex-edge incidence graph is acyclic. We study the (parameterized complexity of) the Hypergraph Feedback Vertex Set (HFVS) problem: given as input a hypergraph H and an integer k, determine whether H has a feedback vertex set of size at most k. It is easy to see that this problem generalizes the classic Feedback Vertex Set (FVS) problem on graphs. Remarkably, despite the central role of FVS in parameterized algorithms and complexity, the parameterized complexity of a generalization of FVS to hypergraphs has not been studied previously. In this paper, we fill this void. Our main results are as follows

- HFVS is W[2]-hard (as opposed to FVS, which is fixed parameter tractable).
- If the input hypergraph is restricted to a linear hypergraph (no two hyperedges intersect in more than one vertex), HFVS admits a randomized algorithm with running time $2^{\mathcal{O}(k^3 \log k)} n^{\mathcal{O}(1)}$.
- If the input hypergraph is restricted to a d-hypergraph (hyperedges have cardinality at most d), then HFVS admits a deterministic algorithm with running time $d^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$.

The algorithm for linear hypergraphs combines ideas from the randomized algorithm for FVS by Becker et al. [J. Artif. Intell. Res., 2000] with the branching algorithm for Point Line Cover by Langerman and Morin [Discrete & Computational Geometry, 2005].

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

Keywords and phrases feedback vertex sets, hypergraphs, FPT, randomized algorithms

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

It would be an understatement to say that Vertex Cover (VC) and Feedback Vertex Set (FVS) have played a pivotal roles in the development of the field of Parameterized Complexity. Vertex Cover asks if given an undirected graph G and a positive integer k, there exists a set S of k vertices which intersects every edge in G. Feedback Vertex Set asks if given an undirected graph G and a positive integer k, there exists a set S (*called feedback vertex set or in short fvs*) of k vertices which intersects every cycle in G. While there has been no improvement in the parameterized algorithm for VC in the last 14 years [9] (the conference version appeared in MFCS 2006), faster algorithms for FVS have been developed over the last decade. The best known algorithm for VC runs in time $\mathcal{O}(1.2738^k + kn)$ [9]. On the other hand, for FVS, the first deterministic $\mathcal{O}(c^k n^{\mathcal{O}(1)})$ algorithm was designed only in 2005; independently by Dehne et al. [14] and Guo et al. [21]. It is important to note here that a randomized algorithm for FVS with running time $\mathcal{O}(4^k n^{\mathcal{O}(1)})$ [5] was known in as early as 1999. The deterministic algorithms led to the race of improving the base of the exponent for FVS algorithms and several algorithms [6, 7, 8, 12, 22, 26, 28], both deterministic and randomized, have been designed. Until few months ago the best known deterministic algorithm for FVS ran in time $3.619^k n^{\mathcal{O}(1)}$ [26], while the Cut and Count technique by Cygan et al. [12] gave the best known randomized algorithm running in time $3^k n^{\mathcal{O}(1)}$. However, just in last few months both these algorithms have been improved; Iwata and Kobayashi [22, IPEC 2019] designed the fastest known deterministic algorithm with running time $\mathcal{O}(3.460^k n)$ and Li and Nederlof [28, SODA 2020] designed the fastest known randomized algorithm with running time $2.7^k n^{\mathcal{O}(1)}$. We would like to remark that many variants of FVS have been studied in literature such as Connected FVS [12, 32], Independent FVS [2, 29, 31], Simultaneous FVS [4, 35] and Subset FVS [13, 23, 24, 25, 30].

The main objective of this paper is a study of FVS on hypergraphs. A hypergraphs is a set family H with a universe $V(H)$ and a family of hyperedges $E(H)$, where each hyperedge (or edge) is a subset of $V(H)$. If every hyperedge in $E(H)$ is of size at most d, it is known as a d-hypergraph. Observe that if each hyperedge is of size *exactly* two, we get an undirected graph. The natural question is, how does VC generalize to hypergraphs. If (G, k) is an instance of VC, we can view VC as the following problem: Given a hypergraph with vertex set $V(G)$ and the set of hyperedges $E(G)$, does there exist a set of k vertices that intersects every hyperedge. Thus, VC is a special case of Hitting Set (HS): Given a hypergraph H and a positive integer k, does there exist a set of k vertices that intersects every hyperedge. If the size of each hyperedge is upper bounded by d, we refer to the problem as the d-Hitting Set (d-HS) problem. Observe that VC is equivalent to the 2-HS problem. It is well known that HS does not admit an algorithm with running time $f(k)n^{\mathcal{O}(1)}$, where the function f depends only on k due to Exponential Time Hypothesis (ETH). That is, the problem is known to be W[2]-hard. On the other hand, d-HS is solvable in time $d^k n^{\mathcal{O}(1)}$ and admits a kernel of size $\mathcal{O}(k^d)$ [1, 18]. It is worth to note d-HS does not admit a kernel of size $\mathcal{O}(k^{d-\epsilon})$ under plausible complexity theory assumptions [15]. Thus, generalization of VC on hypergraphs is well studied. However, there is very little study of FVS on hypergraphs. The only known algorithmic result is a factor d approximation for FVS on d-hypergraphs [20]. Upper bounds on minimum fvs in 3-uniform linear hypergraphs are studied in [16].

> The objective of this paper is to study the hypergraph variant of the Feedback Vertex Set problem from the view point of Parameterized Complexity.

One of the main reasons for the lack of study of FVS on hypergraphs is that it is not as natural to define the generalization of FVS in hypergraphs, as it is for the case of VC (generalizing to HS and d-HS) in hypergraphs. To generalize the notion of fvs to hypergraphs, we need to have notions of *cycles* and *forests* in hypergraphs. For cycles, we use the same notion as that in graph theory [16]: a cycle in a hypergraph H is a sequence $(v_0, e_0, v_1, \ldots, v_\ell, e_\ell, v_0)$ such that $v_0, \ldots, v_\ell$ are distinct vertices, $e_0, \ldots, e_\ell$ are distinct hyperedges, $\ell \geq 1$ and $v_i, v_{(i+1) \bmod (\ell+1)} \in e_i$ for any $i \in \{0, \ldots, \ell\}$. Given the above definition of cycle, a subset S of vertices in a hypergraph H is called a *feedback vertex set*, if there does not exist a cycle in the hypergraph obtained after *deleting* vertices in S. The next natural question is what do we mean by *deletion* of a vertex in a hypergraph. There are two ways to define the vertex deletion operation in hypergraphs:

1. *Strong deletion* or simply *deletion* of a vertex v implies deleting v along with all the hyperedges containing the vertex v.

2. *Weak deletion* of a vertex v implies deleting v without deleting the hyperedges that contain v. That is, the hypergraph H' obtained after weak deletion of a vertex v from H has vertex set $V(H)$ and edge set $\{e \in E(H) : v \notin e\} \cup \{e \setminus \{v\} : e \in E(H), v \in e, |e| > 2\}$.

For a hypergraph H we use the notation $H - S$ to denote the graph obtained after (weak/strong) deletion of the vertices in S. Consequently, there are two ways one may define the Feedback Vertex Set problem – Weak FVS and Strong FVS.

Our Results and Methods. Given a hypergraph H, the incidence graph G corresponding to H is the bipartite graph with bipartition $V(G) = A \uplus B$ where $A = V(H)$ and $B = E(H)$, and for any $v \in V(H)$ and $e \in E(H)$, ve is an edge in G if and only if $v \in e$ in H. Observe that Weak FVS corresponds to finding a fvs S in G of size at most k, such that $S \subseteq A$ and $G - S$ is a forest. Using the best known algorithm for Weighted FVS [3] running in $3.618^k n^{\mathcal{O}(1)}$ time, we can solve Weak FVS in $3.618^k n^{\mathcal{O}(1)}$ time, by transforming the problem to Weighted FVS. To transform Weak FVS to Weighted FVS we assign every vertex in B a weight of $k+1$, every vertex in A a weight of 1. Now the problem of finding an fvs of weight at most k will be equivalent to solving Weak FVS for the original hypergraph. Thus Weak FVS is not challenging as a parameterized problem.

Hence, we only consider FVS on hypergraphs with respect to *strong deletion*. In particular, we study Hypergraph Feedback Vertex Set (HFVS). Here, given an n-vertex hypergraph H and a positive integer k, the objective is to check whether there exists a set $S \subseteq V(H)$ of size at most k, such that $H - S$ is acyclic. As in the case of HS, it is expected that HFVS is W[2]-hard and this can be proven using a parameter preserving reduction from Set Cover (which is "equivalent" to HS). We prove the following theorem in Section D.

▶ **Theorem 1** (♣[1]). HFVS *is* W[2]-hard *when parameterized by* k.

Theorem 1 is not surprising as a generalization of even VC to hypergraphs i.e. HS, is W[2]-hard.

FVS is a deeply studied problem in Parameterized Complexity, and thus, we tried to generalize the existing algorithms as much as possible. However, considering the problem on general hypergraphs is pushing it too far (Theorem 1). This motivated us to look for families of hypergraphs, which are a strict generalizations of graphs and where FVS turns out to be tractable. Specifically, we study the problem for the cases when the input is restricted to *linear hypergraphs* and *d-hypergraphs*.

A hypergraph H is linear if $|e \cap e'| \leq 1$ for any two distinct hyperedges $e, e' \in E(H)$. We show that for both these families, HFVS admits fixed parameter tractable (FPT) algorithms. Our main result is a randomized algorithm for the case when the input hypergraph is linear, and the size of the hyperedges is not bounded. Thus our positive results are the following.

▶ **Theorem 2** (♣). *There exists a deterministic algorithm for* HFVS *on d-hypergraphs, running in time* $d^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$.

▶ **Theorem 3.** *There exists an* $\mathcal{O}^\star(2^{\mathcal{O}(k^3 \log k)})$ *time*[2] *randomized algorithm for* HFVS *on linear hypergraphs, which produces a false negative output with probability at most* $\frac{1}{n^{\mathcal{O}(1)}}$, *and no false positive output.*

[1] Proofs of results marked with ♣ can be found in the appendix.
[2] Polynomial dependency on n is hidden in $\mathcal{O}^\star$ notation.

The restriction to linear hypergraphs corresponds to exclusion of C_4 or $K_{2,2}$ in the corresponding incidence graph. $K_{i,j}$ refers to the complete bipartite graph with partitions of sizes i and j. There has been extensive work on Red-Blue Dominating Set for $K_{i,j}$ free graphs [11, 19, 33, 34]. Theorem 3 can be viewed as an analog of Red-Blue Dominating Set results for $K_{2,2}$ free graphs.

The starting point of both the above mentioned algorithms (Theorems 2 and 3) is recasting HFVS as an appropriate problem on the incidence graph G of the given hypergraph H. Proof of Theorem 3 starts with the observation that for any subset $S \subseteq V(H)$, $H - S$ is acyclic if and only if $G - N_G[S]$ (notations defined in Section 2) is acyclic. Consequently, HFVS is same as the following problem (see Lemma 17 in appendix for proof).

> Dominating FVS on Bipartite Graphs (DFVSB) **Parameter:** k
> **Input:** A bipartite graph G with bipartition $V(G) = A \uplus B$ and $k \in \mathbb{N}$.
> **Question:** Is there a subset $S \subseteq A$ of size at most k such that $G - N_G[S]$ is acyclic?

For a bipartite graph $G = (A \uplus B, E)$, we say that a subset $S \subseteq A$ is a *dominating feedback vertex set* (dfvs) for G if $G - N[S]$ is acyclic. Let G be the incidence graph of a hypergraph H. Then, notice that H is a d-hypergraph if and only if $\max_{e \in E(H)} d_G(e) \leq d$. Also, H is linear if and only if G is C_4-free. As a result HFVS on d-hypergraphs and linear hypergraphs are equivalent to DFVSB on bipartite graphs $G = (A \uplus B, E)$ with $\max_{w \in B} d(w) \leq d$ and on C_4-free bipartite graphs, respectively.

Theorem 2 shows that for d-hypergraphs, HFVS is similar to d-HS. Proof of Theorem 2 utilizes iterative compression. The compression step involves a branching strategy that uses a measure more generalized than the one used in known FVS algorithms for undirected graphs.

Our proof for Theorem 3 is inspired by the randomized algorithm of Becker et al. [5] that runs in $\mathcal{O}(4^k n^{\mathcal{O}(1)})$ time and the branching algorithm for Point Line Cover by Langerman and Morin [27]. The algorithm of Becker et al. [5] first preprocesses the input graph and transforms it into a graph with minimum degree at least 3 and then shows that for any fvs, at least half the edges in a preprocessed graph are incident to the vertex set of the fvs. This immediately gives the following algorithm: "*pick an edge uniformly at random, then pick a vertex that is an endpoint of this edge uniformly at random and add it to a solution, and recurse*". Let G be the incidence graph of a hypergraph H. First we preprocess G and show that in the preprocessed graph (say G) for any dfvs S of size at most k, at least $1/\mathsf{poly}(k)$ fraction of all the edges are incident to $N[S]$. Here, poly denotes a polynomial function. We call this property α-covering, with α being $\mathsf{poly}(k)$. Let S be a fixed fvs of size at most k. We now compute the probability of finding S. Note that if we randomly pick an edge f (that is, pick an edge from graph G uniformly at random and then select f as the hyperedge incident to the selected edge), then with probability $1/\mathsf{poly}(k)$ there exists a vertex incident to f that is contained in S. However, unlike the case of FVS in graphs, here we cannot randomly select a vertex from f, as the size of f could be independent of k. However, for now let us assume that we can preprocess $G - f$ such that the α-covering property holds even after we delete f from G. We assume that α-covering property holds recursively after each iteration of preprocessing. Suppose we do this process $k^2 + 1$ times. Then we have a collection of hyperedges $\mathcal{F} = \{f_1, \ldots, f_{k^2+1}\}$ such that each of them has a non-trivial intersection with S. Observe that the pairwise intersection of these hyperedges cannot be more than one, since G excludes C_4 as a subgraph (H being a linear hypergraph). However, S is a solution of size at most k, and hence there exist $k+1$ hyperedges $f'_1, \ldots, f'_{k+1}$ in $\mathcal{F}$ such that $|f'_i \cap f'_j| = \{v\}$, $i \neq j$ for some $v \in A = V(H)$. This implies that v must belong to S, as each of $f'_1, \ldots, f'_{k+1}$ has a non-trivial intersection with S and if we don't pick v, then every solution is of size at least $k+1$. Hence, we delete v along with all those edges in H that v participates in, and recursively find a solution of size $k-1$ in the reduced hypergraph.

However, unlike the case with FVS for graphs, in HFVS we cannot delete degree 1 vertices or contract degree 2 vertices directly. When we delete a hyperedge, we need to *remember* that we are seeking a solution that is a dfvs as well as a hitting set for the selected set. To implement this idea in our algorithm, we maintain a family $\mathcal{F}$ such that our solution is a dfvs for G as well as a hitting set for $\mathcal{F}$. We exploit the fact that $|\mathcal{F}| \leq k^2+1$ and design reduction rules to get rid of certain degree 1 vertices and shorten degree 2 paths, as well as caterpillars (defined later) like degree 2 paths. We can show that after these reduction rules are performed, the α-covering property holds for the preprocessed graph, α being $\mathsf{poly}(k)$.

2 Preliminaries

For a positive integer $\ell \in \mathbb{N}$, we use $[\ell]$ to denote the set $\{1,2,\ldots,\ell\}$. We use the term graph to denote a simple graph without multiple edges, loops and labels. For the notations related to graphs that are not explicitly stated here, we refer to the book [17]. For a graph G and a subset of vertices $U \subseteq V(G)$, $N_G(U)$ and $N_G[U]$ denote the open neighborhood and closed neighborhood of U, respectively. That is, $N_G(U) = \{v \in V(G) \colon u \in U \text{ and } uv \in E(G)\} \setminus U$ and $N_G[U] = N_G(U) \cup U$. If $U = \{u\}$, then we write $N_G(u) = N_G(U)$ and $N_G[u] = N_G[U]$. Also, we omit the subscript G, if the graph in consideration is clear from the context. For a graph G, a vertex subset $X \subseteq V(G)$, and an edge subset $F \subseteq E(G)$, we use $G[X]$, $G-X$, and $G-F$ to denote the graph induced by X, the graph induced by $V(G)\setminus X$, and the graph with vertex set $V(G)$ and edge set $E(G)\setminus F$, respectively. Moreover, if $X=\{v\}$, then we write $G-v = G-X$. For a graph G, $X,Y \subseteq V(G)$, and $X \cap Y = \emptyset$, $E(X,Y) \subseteq E(G)$ denotes the set of edges in G whose one endpoint is in X and the other one is in Y. For a graph G and a non-edge uv in G, we use $G+uv$ to denote the graph with vertex set $V(G)$ and edge set $E(G) \cup \{uv\}$. A path P in a graph G is a sequence of distinct vertices $u_1 \ldots u_\ell$ such that for all $i \in [\ell-1]$, $u_i u_{i+1} \in E(G)$. We say that a path $P = u_1 \ldots u_\ell$ in a graph G is a *degree two path* in G, if for each $i \in [\ell]$, the degree of u_i in G, denoted by $d_G(u_i)$, is equal to 2. For a path/cycle P, we use $V(P)$ to denote the set of vertices present in P. A triangle is a cycle consisting of exactly 3 edges. A bipartite graph $G = (A \uplus B, E)$ is called a d-bipartite graph if $d_G(b) \leq d$ for all $b \in B$. For two hypergraphs H_1 and H_2, $H_1 \cup H_2$ denotes the hypergraph with the vertex set $V(H_1) \cup V(H_2)$ and the edge set $E(H_1) \cup E(H_2)$.

3 Feedback Vertex Sets on Linear Hypergraphs

In this section we design an FPT algorithm for HFVS on linear hypergraphs. Towards this, we prove the following result about DFVSB, from which Theorem 3 follows as a corollary.

▶ **Theorem 4.** *There exists an* $\mathcal{O}^\star(2^{\mathcal{O}(k^3 \log k)})$ *time randomized algorithm for* DFVSB *on* C_4*-free bipartite graphs, which produces a false negative output with probability at most* $\frac{1}{n^{\mathcal{O}(1)}}$*, and no false positive output.*

To prove Theorem 4, we first define few generalizations of these problems that appear naturally in the recursive steps. Let $\mathcal{F}$ be a family of sets over a universe A, then we define a bipartite graph $G_\mathcal{F}$ as follows. Let the bipartition of $V(G_\mathcal{F})$ be $A_\mathcal{F} \uplus B_\mathcal{F}$, where $A_\mathcal{F} = A$ and $B_\mathcal{F} = \mathcal{F}$. Edge set $E(G_\mathcal{F}) = \{\{u,Y\} \colon u \in A, u \in Y \in \mathcal{F}\}$. Let G be a C_4 free bipartite graph with bipartition $V(G) = A \uplus B$, and $\mathcal{F}$ be a family of sets over the universe A. We define the graph $G \cup G_\mathcal{F} = (A^* \uplus B^*, E^*)$ as follows. Let $A^* = A$, $B^* = B \uplus B_\mathcal{F}$ and $E^* = E(G) \cup E(G_\mathcal{F})$. The following problem generalizes HFVS on linear hypergraphs.

> Hitting Hypergraph Feedback Vertex Set (HHFVS) **Parameter:** $k + |E(H_2)|$
> **Input:** Two linear hypergraphs H_1,H_2 such that $V(H_1) = V(H_2)$, $E(H_1) \cap E(H_2) = \emptyset$, and $H_1 \cup H_2$ is a linear hypergraph, $k \in \mathbb{N}$.
> **Question:** Does there exist a set $S \subseteq V(H_1)$ of size at most k, such that $H_1 - S$ is acyclic and S is a hitting set for $E(H_2)$?

Observe that, if $H_2 = \emptyset$, HHFVS is the same as HFVS (for linear hypergraphs). Next, we define the "graph" version of HHFVS, which generalizes DFVSB on C_4-free graphs.

> Hitting Dominating Bipartite FVS (HDBFVS) **Parameter:** $k + |\mathcal{F}|$
> **Input:** A C_4 free bipartite graph G with bipartition $V(G) = A \uplus B$, a family $\mathcal{F}$ of subsets of A such that the graph $G \cup G_{\mathcal{F}}$ is a C_4 free bipartite graph, $k \in \mathbb{N}$.
> **Question:** Does there exist a set $S \subseteq A$ of size at most k, such that $G - N[S]$ is a forest and S is a hitting set for $\mathcal{F}$?

We say that an instance $(G = (A \uplus B, E), \mathcal{F}, k)$ is a *valid instance* of HDBFVS, if $\mathcal{F}$ is a family of subsets of A such that the graph $G \cup G_{\mathcal{F}}$ is a C_4-free bipartite graph.

> In the rest of the section, whenever we say $\mathcal{I} = (G = (A \uplus B, E), \mathcal{F}, k)$ is an instance of HDBFVS, it implies that $\mathcal{I}$ is a valid instance of HDBFVS. Further, after each application of a reduction rule, we ensure that the instance remains valid.

The proof of the following simple observation follows from the fact that $G \cup G_{\mathcal{F}}$ is C_4-free.

▷ Observation 3.1. If $(G = (A \uplus B, E), \mathcal{F}, k)$ is an instance of HDBFVS, then (i) pairwise intersection of sets in $\mathcal{F}$ is of size at most 1, and (ii) for every vertex $b \in B$ and $F \in \mathcal{F}$, $|N(b) \cap F|$ is at most one.

Given an instance (H_1, H_2, k) of HHFVS, we can obtain an instance, $(G, \mathcal{F}, k)$, of HDBFVS in a canonical way. Next lemma shows their equivalence.

▶ **Lemma 5 (♣).** *(H_1, H_2, k) is a* YES*-instance of* HHFVS *if and only if $(G, \mathcal{F} = E(H_2), k)$ is a* YES*-instance of* HDBFVS*, where G is the incidence graph of the hypergraph H_1.*

The rest of the section is devoted to designing an FPT algorithm for HDBFVS. Given an instance $(G = (A \uplus B, E), \mathcal{F}, k)$ of HDBFVS, we first define some notations. For a vertex $v \in A$, X_v denotes the set $\{Y \mid Y \in \mathcal{F}, v \in Y\}$. We *distinguish* the vertices in A as follows.

- If $|X_v| \geq 2$, i.e., v is in at least two sets in $\mathcal{F}$, then we say that v is a *special* vertex.
- If $|X_v| = 1$, i.e., v is in exactly one set in $\mathcal{F}$, then we say that v is an *easy* vertex.
- Otherwise, we say that v is a *trivial* vertex.

Let $V(\mathcal{F}) = \{v \in A \mid v \in Y \text{ where } Y \in \mathcal{F}\}$. For a graph $G^\star$, the notations $V_0(G^\star)$, $V_{=1}(G^\star)$, $V_{=2}(G^\star)$, and $V_{\geq 3}(G^\star)$ denote the set of isolated vertices, the set of vertices of degree 1, the set of vertices of degree 2, and the set of vertices of degree at least 3 in $G^\star$, respectively.

▶ **Lemma 6.** *Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of* HDBFVS*. Then, the number of special vertices in A is upper bounded by $\binom{|\mathcal{F}|}{2}$.*

Proof. For contradiction, assume that the number of special vertices in A is more than $\binom{|\mathcal{F}|}{2}$. By pigeonhole principle there exist two special vertices $u, v \in A$, such that $|X_u \cap X_v| \geq 2$. Let $Y_1, Y_2 \in X_u \cap X_v$. This implies that $u, v \in Y_1 \cap Y_2$, contradicting Observation 3.1(i). ◀

Now we state some reduction rules that are applied exhaustively by the algorithm in the order in which they appear. Let $(G, \mathcal{F}, k)$ be an instance of HDBFVS and $(G', \mathcal{F}', k)$ be the resultant instance after application of a reduction rule. To show that a reduction rule is safe, we will prove that $(G, \mathcal{F}, k)$ is a YES-instance if and only if $(G', \mathcal{F}', k)$ is a YES-instance.

▷ Reduction Rule 3.1. If one of the following holds, then return a trivial NO-instance: (i) $k < 0$; (ii) $k = 0$ and G is not acyclic; and (ii) $k = 0$ and $\mathcal{F}$ is not empty.

▷ Reduction Rule 3.2. If $k \geq 0$, G is acyclic and $\mathcal{F}$ is empty, then return a trivial YES-instance.

▷ Reduction Rule 3.3. Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and $b \in B$ be a vertex that does not participate in any cycle in G. Then, output $(G - b, \mathcal{F}, k)$.

▷ Reduction Rule 3.4. Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and $v \in A$ be an isolated vertex in G. If v is a trivial vertex, then output $(G - v, \mathcal{F}, k)$.

It is easy to see that the above reduction rules are safe and can be applied in polynomial time. Observe that, when Reduction Rules 3.3 and 3.4 are no longer applicable, then $V_0(G) \subseteq A$ and each isolated vertex in G is either easy or special. Next, we state a reduction rule that will help to bound the number of easy isolated vertices in G.

▷ Reduction Rule 3.5 ($\star$[3]). Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and $v \in A$ be an isolated vertex in G. Suppose v is an easy vertex, $X_v = \{Y\}$, and $|Y| > 1$. Then output $(G', \mathcal{F}', k)$, where $G' = G - v$ and $\mathcal{F}' = (\mathcal{F} \setminus \{Y\}) \cup \{(Y \setminus \{v\})\}$.

▷ Reduction Rule 3.6 ($\star$). Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and $v \in A$ be a vertex of degree 1 in G. If v is a trivial vertex, then output $(G' = G - v, \mathcal{F}, k)$.

Observe that when Reduction Rules 3.1 to 3.6 are no longer applicable, the following holds.

▶ **Lemma 7.** *Let $(G, \mathcal{F}, k)$ be an instance reduced with respect to Reduction Rules 3.1 to 3.6. Then, the following holds.*

1. $V_0(G) \cup V_{=1}(G) \subseteq A$*, all vertices in* $V_0(G) \cup V_{=1}(G)$ *are either easy or special.*
2. $|V_0(G)| \leq |\mathcal{F}| + \binom{|\mathcal{F}|}{2}$.

▶ **Lemma 8.** *For any vertex $b \in B$, $|N_G(b) \cap V_{=1}(G)| \leq |\mathcal{F}|$.*

Proof. If there exists a vertex $v \in N_G(b) \cap V_{=1}(G)$ which is a trivial vertex, then Reduction Rule 3.6 is applicable. Thus, (i) for all $v \in N_G(b) \cap V_{=1}(G)$, v belongs to some set in $\mathcal{F}$. For contradiction, let $b \in B$ be a vertex such that $N_G(b)$ contains at least $|\mathcal{F}| + 1$ vertices of degree 1 in G. Then, by pigeonhole principle and statement (i), at least two degree 1 vertices say $u, v \in N_G(b)$ are contained in a set $Y \in \mathcal{F}$, which is a contradiction to item (ii) of Observation 3.1. This completes the proof of the lemma. ◀

Recall that, P is a degree two path in G if each vertex in P has degree exactly two in G. Next we state the reduction rules that help us bound the length of long degree two paths in $G - V_{=1}(G)$, i.e., to bound the length of degree two paths in the graph obtained after deleting vertices of degree 1 from G. Towards this, we first define the notion of a *nice path*.

▶ **Definition 9.** *We say that P is a* nice path *in G, if P does not have any* special *vertex and the degree of each vertex in P in the graph $G - V_{=1}(G)$ is exactly 2. A* nice path *P in G is a* degree two nice path *if each vertex in P has degree exactly 2 in G.*

[3] The safeness proofs of reduction rules marked with $\star$ are moved to Section B in the appendix.

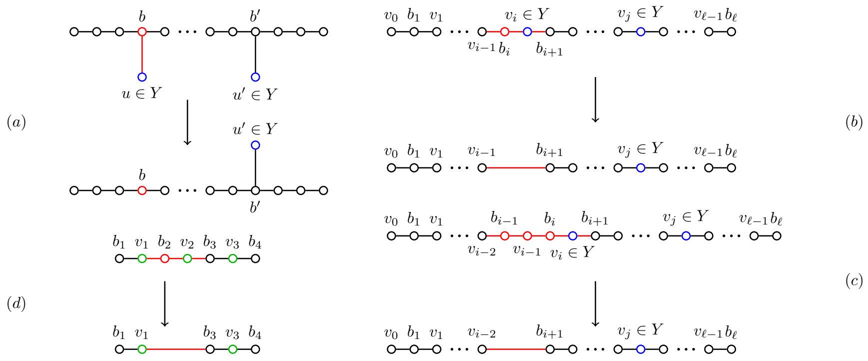

Figure 1 (a) is an illustration of Reduction Rule 3.7, (b) and (c) are illustrations of two cases of Reduction Rule 3.8, (d) is an illustration of Reduction Rule 3.9. In (a), (b) and (c) *blue* vertices denote *easy* vertices, and in (d) *green* vertices denote *trivial* vertices.

▷ Reduction Rule 3.7 ($\star$). Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS, P be a nice path in G and $b, b' \in B$ be two vertices in P. If there exist two easy vertices u, u' whose degree is 1 in G, adjacent to b, b', respectively, such that $X_u = X_{u'} = \{Y\}$, then return $(G', \mathcal{F}', k)$, where $G' = G - u$, $\mathcal{F}' = (\mathcal{F} \setminus \{Y\}) \cup \{Y \setminus \{u\}\}$.

▶ **Lemma 10.** *Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of* HDBFVS *reduced with respect to Reduction Rules 3.1 to 3.7. Then, in any nice path P in G, the number of vertices that are adjacent to a vertex of degree 1 in G is bounded by $\binom{|\mathcal{F}|}{2} + |\mathcal{F}|$.*

Proof. From statement 1 in Lemma 7, we have that $V_{=1}(G) \subseteq A$. This implies, $N_G(V_{=1}(G)) \subseteq B$. Also, each vertex in $V_{=1}(G)$ is either easy or special. By Lemma 6, the number of vertices that are special is bounded by $\binom{|\mathcal{F}|}{2}$. Therefore, the number of vertices in P that are adjacent to special degree 1 vertices is at most $\binom{|\mathcal{F}|}{2}$. Since Reduction Rule 3.7 is no longer applicable, we have that corresponding to each set $Y \in \mathcal{F}$, there exists at most 1 vertex in P that has a degree 1 neighbor u such that $X_u = \{Y\}$. This implies that at most $|\mathcal{F}|$ vertices in P can be adjacent to degree 1 easy vertices, resulting in the mentioned upper bound. ◀

The next reduction rule helps us in upper bounding the length of degree two paths in G.

▷ Reduction Rule 3.8 ($\star$). Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and $P = v_0 b_1 v_1 \ldots v_{\ell-1} b_\ell$ be a degree two nice path in G, where $\{b_1, \ldots, b_\ell\} \subseteq B$, $\{v_0, \ldots, v_{\ell-1}\} \subseteq A$, and $\ell \geq 5$. Let $v_i, v_j \in A \cap (V(P) \setminus \{v_0, v_1\})$ be two distinct easy vertices such that $X_{v_i} = X_{v_j} = \{Y\}$ for some $Y \in \mathcal{F}$ and $i < j$. Then, return $(G', \mathcal{F}', k)$, where G' and $\mathcal{F}'$ are defined as follows.

- If $X_{v_{i-1}} \neq X_{v_{i+1}}$ or $X_{v_{i-1}} = X_{v_{i+1}} = \emptyset$, then let $G' = (G - \{b_i, v_i\}) + v_{i-1} b_{i+1}$ (i.e., G' be the graph obtained by deleting the vertices b_i, v_i from G and by adding a new edge $v_{i-1} b_{i+1}$) and $\mathcal{F}' = (\mathcal{F} \setminus \{Y\}) \cup \{Y \setminus \{v_i\}\}$.
- Otherwise, $X_{v_{i-1}} = X_{v_{i+1}} = \{Y^\star\}$, then let $G' = (G - \{b_{i-1}, v_{i-1}, b_i, v_i\}) + v_{i-2} b_{i+1}$ (i.e., G' be the graph obtained by deleting the vertices $b_{i-1}, v_{i-1}, b_i, v_i$ from G and by adding a new edge $v_{i-1} b_{i+1}$) and $\mathcal{F}' = (\mathcal{F} \setminus \{Y, Y^\star\}) \cup \{Y^\star \setminus \{v_{i-1}\}, Y \setminus \{v_i\}\}$.

Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS reduced with respect to Reduction Rules 3.1 to 3.8. Observe that, for each set $Y \in \mathcal{F}$ and a degree two nice path P in G, the number of easy vertices among the last $|V(P)| - 3$ vertices in $V(P)$ that belong to Y, is upper bounded by one. Reduction Rule 3.8 leads us to the following observation.

▷ Observation 3.2. Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be a reduced instance of HDBFVS with respect to Reduction Rules 3.1 to 3.8. Then, in any degree two nice path P of length at least 10 in G, the number of easy vertices is bounded by $|\mathcal{F}| + 2$.

▷ Reduction Rule 3.9 ($\star$). Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and $P = b_1v_1b_2v_2b_3v_3b_4$ be a degree two nice path in G, such that $\{b_1, \ldots, b_4\} \subseteq B$, $\{v_1, v_2, v_3\} \subseteq A$ and v_1, v_2, v_3 are trivial vertices. Then, return $(G', \mathcal{F}, k)$, where G' is the graph obtained by deleting the vertices b_2, v_2 from G and adding a new edge v_1b_3 (i.e., $G' = (G - \{v_2, b_2\}) + v_1b_3$).

▷ Observation 3.3. Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of HDBFVS and let $(G' = (A' \uplus B', E'), \mathcal{F}', k')$ be the reduced instance of HDBFVS obtained from $(G = (A \uplus B, E), \mathcal{F}, k)$, by exhaustive applications of Reduction Rules 3.1 to 3.9. Then, $|\mathcal{F}'| = |\mathcal{F}|$ and $k' \leq k$.

We now bound the size of degree 2 path, when there is no degree 1 vertex in the graph.

▶ **Lemma 11 (♣).** *Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an instance of* HDBFVS *reduced with respect to Reduction Rules 3.1 to 3.9. Then, the number of vertices in a degree two path P in $G - V_{=1}(G)$ is bounded by $63|\mathcal{F}|^5 + 21$.*

From now on, we say that $(G = (A \uplus B, E), \mathcal{F}, k)$ is a *reduced instance* of HDBFVS if it is reduced with respect to Reduction Rules 3.1 to 3.9. In the following lemma, we observe that, if $(G = (A \uplus B, E), \mathcal{F}, k)$ is a YES-instance of HDBFVS, then a large number of edges in G is incident to the neighborhood of the solution.

▶ **Lemma 12.** *Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be a reduced instance of* HDBFVS *where G is not a forest. Then, for any solution S, at least $1/(445|\mathcal{F}|^6 + 68)$ fraction of the total edges in E are incident to $N[S]$.*

Proof. Let E_S be the set of edges incident to all the vertices of $N[S]$ in G. Observe that, $E(G) = E_S \uplus E(G - N[S])$. Since $G - N[S]$ is a forest, we have that $|E(G - N[S])| < |V(G - (N[S] \cup V_0(G)))|$. We aim to show that $|V(G - (N[S] \cup V_0(G)))| \leq (445|\mathcal{F}|^6 + 67) \cdot |E_S|$. Let $V^\star$ be the set of vertices of degree 1 in $G - N[S]$. Let $V_1^\star \subseteq V^\star$ be the set of vertices that have some neighbor in $N[S]$ and $V_2^\star = V^\star \setminus V_1^\star$. That is, $V_2^\star \subseteq V_{=1}(G)$. Since the vertices in $V_1^\star$ have neighbors in $N[S]$, they contribute at least one edge to the set E_S and these edges are distinct. Hence, $|V_1^\star| \leq |E_S|$.

Since $V_2^\star \subseteq V_{=1}(G)$, by Lemma 7, we have that $V_2^\star \subseteq A$. Thus, $V_2^\star$ have neighbors only in the set $B \cap V(G - N[S])$. Also, by Lemma 8, any vertex in B can be adjacent to at most $|\mathcal{F}|$ vertices of degree 1 in G. Hence, each vertex in $B \cap V(G - N[S])$ can be adjacent to at most $|\mathcal{F}|$ vertices of $V_2^\star$. Thus, we have that $|V_2^\star| \leq |\mathcal{F}| \cdot |B \cap V(G - N[S])|$. Let G' be the graph $G - (V_0(G) \cup V_2^\star)$. Since $V_0(G) \cup V_2^\star \subseteq A$, we have that, $B \subseteq V(G')$ and $B \cap V(G - N[S]) = B \cap V(G' - N[S])$. Hence, we obtain the following.

$$|V_2^\star| \leq |\mathcal{F}| \cdot |B \cap V(G' - N[S])| \leq |\mathcal{F}| \cdot |V(G' - N[S])| \quad (1)$$

$$|V^\star| = |V_1^\star| + |V_2^\star| \leq |\mathcal{F}| \cdot |V(G' - N[S])| + |E_S| \quad \text{(By (1) and } |V_1^\star| \leq |E_S|) \quad (2)$$

Since the graph G' is obtained from G by deleting a subset of vertices that are contained in $V_0(G) \cup V_{=1}(G) \subseteq A$, the vertices that are degree 1 in $G' - N[S]$ are either degree 1 vertices in $G - N[S]$ and are contained in A, in particular in $V_1^\star$, or they are contained in B and are neighbors of vertices in $V_2^\star$ in G. Let L be the set of leaves (vertices of degree 1) in $G' - N[S]$. We claim that $L = V_1^\star$. For contradiction, assume that a vertex $b \in B \cap L$. Since Reduction Rule 3.3 is no longer applicable, we have that each vertex in B participates in a cycle in G and hence, participates in a cycle in G'. Therefore, degree of b is at least 2 in G'. Observe that b cannot have a neighbor in S, otherwise $b \in N[S]$. This implies that b has 2 neighbors in $G' - N[S]$, which contradicts that $b \in L$. Observe that each vertex in $V_1^\star$ is a leaf vertex in $G' - N[S]$. Hence $L = V_1^\star$. Therefore, we obtain the following.

$$\begin{aligned} |L| &\leq |E_S|. && (3)\\ V_{\geq 3}(G' - N[S]) &\leq |E_S| && \text{(Since, } G' - N[S] \text{ is a forest, } V_{\geq 3}(G' - N[S]) \leq |L|) \quad (4) \end{aligned}$$

Next we bound $|V_0(G' - N[S])|$. Since, for any vertex v in $G' - N[S]$, $d_G(v) \geq 1$, we have that any vertex $w \in V_0(G' - N[S])$ is adjacent to some vertex in $N[S]$. Then, each vertex in $V_0(G' - N[S])$ contributes at least 1 edge to the set E_S and these edges are distinct.

$$\text{Therefore,} \qquad |V_0(G' - N[S])| \leq |E_S|. \tag{5}$$

Let $V^1_{=2}(G')$ be the set of vertices of degree 2 in $G' - N[S]$ that have a neighbor in $N[S]$. Then, each vertex in $V^1_{=2}(G')$ contributes at least 1 edge to the set E_S. Therefore, we have

$$|V^1_{=2}(G')| \leq |E_S|. \tag{6}$$

Let $V^2_{=2}(G')$ be the set of vertices of degree 2 in $G' - N[S]$, that do not have a neighbor in $N[S]$. Then, each vertex in $V^2_{=2}(G')$ is contained in some maximal degree two path not containing any vertex of $V^1_{=2}(G')$ in $G' - N[S]$. Observe that, since $G' - N[S]$ is a forest, (i) the number of maximal degree two paths not containing any vertex of $V^1_{=2}(G')$ in $G' - N[S]$ is bounded by $|L \cup V_{\geq 3}(G') \cup V^1_{=2}(G')|$ and hence bounded by $3|E_S|$ (because of (3),(4), and (6)). Observe that a degree two path not containing any vertex of $V^1_{=2}(G')$ in $G' - N[S]$ is also a degree two path in $G - V_{=1}(G)$. By Lemma 11, (ii) the number of vertices in a degree two path in $G - V_{=1}(G)$ is bounded by $63|\mathcal{F}|^5 + 21$. So, statements (i) and (ii) imply that

$$|V^2_{=2}(G')| \leq (189|\mathcal{F}|^5 + 63)|E_S| \tag{7}$$

Observe that $V_{=2}(G' - N[S]) = V^1_{=2}(G') \cup V^2_{=2}(G')$. By (6) and (7), we get the following.

$$|V_{=2}(G' - N[S])| = |V^1_{=2}(G')| + |V^2_{=2}(G')| \leq (189|\mathcal{F}|^5 + 64)|E_S| \tag{8}$$

Note that, $V(G' - N[S]) = V_0(G' - N[S]) \cup L \cup V_{\geq 3}(G' - N[S]) \cup V_{=2}(G' - N[S])$. Hence, we obtain the following using (3), (5), (4), and (8).

$$\begin{aligned} |V(G' - N[S])| &= |V_0(G' - N[S])| + |L| + |V_{\geq 3}(G' - N[S])| + |V_{=2}(G' - N[S])| \\ &\leq |E_S| + |E_S| + |E_S| + (189|\mathcal{F}|^5 + 64)|E_S| \\ &\leq (189|\mathcal{F}|^5 + 67)|E_S| && (9) \end{aligned}$$

Using (1) and (9), we obtain the following.

$$\begin{aligned} |V(G - (N[S] \cup V_0(G)))| &\leq |V(G' - N[S])| + |V_2^\star| \\ &\leq (|\mathcal{F}| + 1)|V(G' - N[S])| \qquad \text{(By (1))} \\ &\leq (|\mathcal{F}| + 1)((189|\mathcal{F}|^5 + 67)|E_S|) \\ &\leq (445|\mathcal{F}|^6 + 67)|E_S| \\ \text{Thus,} \qquad |E(G)| &= |E_S| + |E(G - N[S])| \\ &\leq |E_S| + |V(G - (N[S] \cup V_0(G))| \leq (445|\mathcal{F}|^6 + 68)|E_S|. \end{aligned}$$

This concludes the proof. ◀

▶ **Lemma 13.** *Let* $(G = (A \uplus B, E), \mathcal{F}, k)$ *be an instance of* HDBFVS*, where* G *is a forest and* $|\mathcal{F}| \leq k^2$*. Then, there exists an algorithm which solves the instance in* $\mathcal{O}^\star((2k^4)^k)$ *time.*

Proof. The Algorithm first applies Reduction Rules 3.1 to 3.9 exhaustively in the order in which they are stated. If any reduction rule solves the instance, then output YES and NO accordingly. All the reduction rules are safe, and can be applied in polynomial time, and they can be applied only polynomial many times since each reduction rule decreases the size of the graph. Let $(G' = (A' \uplus B', E'), \mathcal{F}', k')$ be the reduced instance. Since Reduction Rule 3.3 is no longer applicable, $B' = \emptyset$, and hence G' is an edge-less graph with vertex set A'. By Lemma 7, $|V(G')| = |A'| \leq |\mathcal{F}'| + \binom{|\mathcal{F}'|}{2}$. By Observation 3.3, we have that $|\mathcal{F}'| = |\mathcal{F}| \leq k^2$ and hence, $|V(G')| \leq 2k^4$. We enumerate all the subsets of $V(G')$ of size at most k and check if they forms a solution; else return a NO-instance. The algorithm runs in time $\binom{2k^4}{k} n^{\mathcal{O}(1)} = \mathcal{O}^\star((2k^4)^k)$. This completes the proof. ◀

▶ **Lemma 14.** *There is a randomized algorithm that takes an instance* $(G = (A \uplus B, E), \mathcal{F}, k)$ *of* HDBFVS *as input, runs in* $\mathcal{O}^\star((2k^4)^k)$ *time, and outputs either* YES*, or* NO*, or an instance* $(G^\star = (A^\star \uplus B^\star, E^\star), \mathcal{F}^\star, k^\star)$ *of* HDBFVS *where* $k^\star < k$*, with the following guarantee.*

- *If* $(G, \mathcal{F}, k)$ *is a* YES*-instance, then the output is* YES *or an equivalent* YES*-instance* $(G^\star, \mathcal{F}^\star, k^\star)$ *where* $k^\star < k$*, with probability at least* $(445k^{12} + 68)^{-(k^2+1)}$.
- *If* $(G, \mathcal{F}, k)$ *is a* NO*-instance, then the output is* NO *or an equivalent* NO*-instance* $(G^\star, \mathcal{F}^\star, k^\star)$ *where* $k^\star < k$*, with probability* 1*.*

Proof. Let $(G = (A \uplus B, E), \mathcal{F}, k)$ be an input instance of HDBFVS. Recall that, for any $v \in A$, $X_v = \{F \in \mathcal{F} : v \in F\}$. The algorithm applies the following iterative procedure.

Step 1. If G is acyclic and $|\mathcal{F}| \leq k^2$, then apply Lemma 13 and solve the instance.

Step 2. If $|\mathcal{F}| \geq k^2 + 1$;
  (i) If there exists a vertex v such that $|X_v| \geq k+1$, return $(G - N[v], \mathcal{F} \setminus X_v, k-1)$.
  (ii) Otherwise, return that $(G = (A \uplus B, E), \mathcal{F}, k)$ is a NO-instance of HDBFVS.

Step 3. Apply Reduction Rules 3.1 to 3.9 exhaustively in the order in which they are stated. If any reduction rule solves the instance, then output YES and NO accordingly. Let $(G' = (A' \uplus B', E'), \mathcal{F}', k')$ be the reduced instance.

Step 4. Pick an edge $e = ub$ in $E(G')$ uniformly at random, where $u \in A', b \in B'$. Set $G := G' - b, \mathcal{F} := \mathcal{F}' \cup \{N_{G'}(b)\}$, and $k := k'$. Go to Step 1.

Now we prove the correctness of the algorithm. Correctness of Step 1 follows from Lemma 13. Next assume that $|\mathcal{F}| \geq k^2 + 1$. Let v be a vertex that is contained in at least $k+1$ sets in $\mathcal{F}$. By Observation 3.1, pairwise intersection of two sets in $\mathcal{F}$ is at most 1. Thus, if we do not pick v in our solution, then we have to pick at least $k+1$ vertices to hit the sets in X_v. Thus v belongs to every solution of $(G, \mathcal{F}, k)$ of HDBFVS. Hence, $(G, \mathcal{F}, k)$ is a YES-instance of HDBFVS if and only if $(G - v, \mathcal{F} \setminus X_v, k-1)$ is a YES-instance of HDBFVS, and correctness of Step 2i follows. Suppose each vertex in A is contained in at most k sets of $\mathcal{F}$. Thus no set of size at most k can hit $k^2 + 1$ sets of $\mathcal{F}$. Hence, $(G, \mathcal{F}, k)$ is a NO-instance of HDBFVS, and correctness of Step 2ii follows. Correctness of the Step 3 is implied by the safeness of reduction rules. Suppose the algorithm does not stop in Step 3. Let $(G', \mathcal{F}', k')$ be the reduced instance, where $k' \leq k$. Now, let S be a hypothetical solution to $(G', \mathcal{F}', k')$. By Lemma 12, the picked edge $e = ub$ is incident to a vertex in $N_{G'}[S]$ with probability at least $1/(445|\mathcal{F}|^6 + 68)$. This implies that with probability at least $1/(445|\mathcal{F}|^6 + 68)$ a vertex in $N_{G'}(b)$ is contained in S. Hence, if $(G', \mathcal{F}', k')$ is a YES-instance, then $(G' - b, \mathcal{F}' \cup \{N_G(b)\}, k')$ is a YES-instance, with probability at least $1/(445|\mathcal{F}|^6 + 68)$. Also, notice that any solution to $(G' - b, \mathcal{F}' \cup \{N_G(b)\}, k')$ is also a solution to $(G', \mathcal{F}', k')$. Hence, if $(G', \mathcal{F}', k')$ is a NO-instance, then $(G' - b, \mathcal{F}' \cup \{N_G(b)\}, k')$ is a NO-instance, with probability 1. Consequently, if $(G, \mathcal{F}, k)$ is a NO-instance, then the output is NO or a NO-instance $(G^\star, \mathcal{F}^\star, k^\star)$ with probability 1.

Let $(G, \mathcal{F}, k)$ be a YES-instance. By Observation 3.3, after the application of Reduction Rules 3.1 to 3.9, in the reduced instance, $|\mathcal{F}'| = |\mathcal{F}|$. Thus, Step 4 is applied at most $k^2 + 1$ times. Each execution of Step 4 is a *success* with probability at least $1/(445|\widehat{\mathcal{F}}|^6 + 68)$, where $\widehat{\mathcal{F}}$ is the family in the instance considered in that step. In Step 4, the size of the family of any instance is bounded by k^2, due to Step 2. Hence each execution of Step 4 is a success with probability at least $1/(445k^{12} + 68)$. This implies that either our algorithm outputs YES or a YES-instance $(G^\star, \mathcal{F}^\star, k^\star)$ with probability at least $(445k^{12} + 68)^{-(k^2+1)}$. By Observation 3.3, we know that after the application of Reduction Rules 3.1 to 3.9, the parameter k' in the reduced instance is at most the parameter k in the original instance. Moreover, if the algorithm outputs an instance, then that will happen in Step 2i and there k decreases by 1. Thus $k^\star < k$. This proves the correctness of the algorithm.

By Lemma 13, Step 1 runs in $\mathcal{O}^\star((2k^4)^k)$ time. Observe that, Step 2 runs in polynomial time. All the reduction rules run in polynomial time, and are applied only polynomially many times. Step 4 runs in polynomial time, and we have at most $k^2 + 1$ iterations. Therefore, the total running time is $\mathcal{O}^\star((2k^4)^k)$. This completes the proof. ◀

By applying Lemma 14 at most k times, we can show the the following.

▶ **Lemma 15.** *There exists a randomized algorithm $\mathcal{B}$ that takes an instance $(G = (A \uplus B, E), \mathcal{F}, k)$ of* HDBFVS *as input, runs in $\mathcal{O}^\star((2k^4)^k)$ time, and outputs either* YES *or* NO *with the following guarantee. If $(G, \mathcal{F}, k)$ is a* YES*-instance, then the output is* YES *with probability at least $(445k^{12} + 68)^{-k(k^2+1)}$. If $(G, \mathcal{F}, k)$ is a* NO*-instance, then the output is* NO *with probability* 1.

Let $\tau(k) = (445k^{12} + 68)^{k(k^2+1)}$. To boost the success probability of algorithm $\mathcal{B}$, we repeat it $\mathcal{O}(\tau(k) \log n)$ times. After applying algorithm $\mathcal{B}$ $\mathcal{O}(\tau(k) \log n)$ times, the success probability is at least $1 - \left(1 - \frac{1}{\tau(k)}\right)^{\mathcal{O}(\tau(k) \log n)} \geq 1 - \frac{1}{2^{\mathcal{O}(\log n)}} \geq 1 - \frac{1}{n^{\mathcal{O}(1)}}$.

Thus, we have the following result.

▶ **Theorem 16.** *There exists a randomized algorithm $\mathcal{A}$ that takes an instance $(G = (A \uplus B, E), \mathcal{F}, k)$ of* HDBFVS *as input, runs in $\mathcal{O}^\star(2^{\mathcal{O}(k^3 \log k)})$ time, and outputs either* YES *or* NO *with the following guarantee.*

- *If $(G, \mathcal{F}, k)$ is a* YES*-instance, then the output is* YES *with probability at least $1 - \frac{1}{n^{\mathcal{O}(1)}}$.*
- *If $(G, \mathcal{F}, k)$ is a* NO*-instance, then the output is* NO *with probability* 1.

4 Conclusion and Open Problems

In this paper, we initiated the study of Feedback Vertex Set problem on hypergraphs. We showed that the problem is W[2]-hard on general hypergraphs. However, when the input is restricted to d-hypergraphs and linear hypergraphs, which are a strict generalization of graphs, FVS turns out to be tractable (FPT). Derandomization of the randomized FVS algorithm given in this paper is yet to be explored. We believe that this opens up a new direction in the study of parameterized algorithms. That is, extending the study of other graph problems, in the realm of Parameterized Complexity, to hypergraphs. Designing substantially faster algorithms for HFVS on linear hypergraphs and designing polynomial kernels remain interesting questions for the future.

Acknowledgements. We thank the anonymous referees of an earlier version of the paper. Their comments helped us a lot in improving the paper.

References

1 Faisal N. Abu-Khzam. A kernelization algorithm for d-hitting set. *J. Comput. Syst. Sci.*, 76(7):524–531, 2010.

2 Akanksha Agrawal, Sushmita Gupta, Saket Saurabh, and Roohani Sharma. Improved algorithms and combinatorial bounds for independent feedback vertex set. In *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63, pages 2:1–2:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

3 Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshtanov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In *LATIN 2016: Theoretical Informatics - 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings*, volume 9644 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2016.

4 Akanksha Agrawal, Daniel Lokshtanov, Amer E. Mouawad, and Saket Saurabh. Simultaneous feedback vertex set: A parameterized perspective. *TOCT*, 10(4):18:1–18:25, 2018.

5 Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *J. Artif. Intell. Res.*, 12:219–234, 2000.

6 Yixin Cao. A naive algorithm for feedback vertex set. In *1st Symposium on Simplicity in Algorithms, SOSA 2018, January 7-10, 2018, New Orleans, LA, USA*, volume 61, pages 1:1–1:9. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

7 Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, 73(1):63–86, 2015.

8 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74(7):1188–1198, 2008.

9 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010.

10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

11 Marek Cygan, Fabrizio Grandoni, and Danny Hermelin. Tight kernel bounds for problems on graphs with small degeneracy. *ACM Trans. Algorithms*, 13(3):43:1–43:22, 2017.

12 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011.

13 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discrete Math.*, 27(1):290–309, 2013.

14 Frank K. H. A. Dehne, Michael R. Fellows, Michael A. Langston, Frances A. Rosamond, and Kim 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.

15 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

16 Zhuo Diao and Zhongzheng Tang. On the feedback number of 3-uniform hypergraph. volume abs/1807.10456, 2018. arXiv:1807.10456.

17 Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, 2012.

18 J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag, 2006.

19 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.

20 Toshihiro Fujito. Approximating minimum feedback vertex sets in hypergraphs. *Theoretical Computer Science*, 246(1):107 – 116, 2000.

21 Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.*, 72(8):1386–1396, 2006.

22 Yoichi Iwata and Yusuke Kobayashi. Improved analysis of highest-degree branching for feedback vertex set. In *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, pages 22:1–22:11, 2019.

23 Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, LP-branching, and FPT algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016.

24 Yoichi Iwata, Yutaro Yamaguchi, and Yuichi Yoshida. 0/1/all CSPs, half-integral a-path packing, and linear-time FPT algorithms. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 462–473. IEEE Computer Society, 2018.

25 Ken-ichi Kawarabayashi and Yusuke Kobayashi. Fixed-parameter tractability for the subset feedback set problem and the s-cycle packing problem. *J. Comb. Theory, Ser. B*, 102(4):1020–1034, 2012.

26 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Inf. Process. Lett.*, 114(10):556–560, 2014.

27 Stefan Langerman and Pat Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005.

28 Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 971–989, 2020.

29 Shaohua Li and Marcin Pilipczuk. An improved FPT algorithm for independent feedback vertex set. In *Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings*, volume 11159, pages 344–355. Springer, 2018.

30 Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. Linear time parameterized algorithms for subset feedback vertex set. *ACM Trans. Algorithms*, 14(1):7:1–7:37, 2018.

31 Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, and Saket Saurabh. On parameterized independent feedback vertex set. *Theor. Comput. Sci.*, 461:65–75, 2012.

32 Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, Saket Saurabh, and Somnath Sikdar. FPT algorithms for connected feedback vertex set. *J. Comb. Optim.*, 24(2):131–146, 2012.

33 Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Trans. Algorithms*, 9(1):11:1–11:23, 2012.

34 Jan Arne Telle and Yngve Villanger. FPT algorithms for domination in sparse graphs and beyond. *Theor. Comput. Sci.*, 770:62–68, 2019.

35 Junjie Ye. A note on finding dual feedback vertex set. *CoRR*, abs/1510.00773, 2015. arXiv:1510.00773.

A Equivalence between HFVS and DFVSB

▶ **Lemma 17.** *(H, k) is a* YES *instance of* HFVS *if and only if $(G = (A \uplus B, E'), k)$ is a* YES *instance of* DFVSB*, where G is the incidence graph of the hypergraph H.*

Proof. In forward direction, let S be a solution to (H, k) of HFVS. We claim that S is also a solution to $(G = (A \uplus B, E'), k)$ of DFVSB. Suppose not. Then, there exists a cycle $C = v_1 e_1 \dots v_\ell e_\ell v_1$ in the graph $G - N_G[S]$. This implies that $e_1, \dots, e_\ell$ are hyperedges in $H - S$, and $\{v_1, \dots, v_\ell\} \subseteq V(H) \setminus S$. Then $(v_1, e_1, \dots, v_\ell, e_\ell, v_1)$ is a cycle in the hypergraph $H - S$. This is a contradiction to the assumption that S is a solution to (H, k).

In reverse direction, let S' be a solution to (G, k) of DFVSB. We claim that S' is also a solution to (H, k) of HFVS. Suppose not. Then, there exists a cycle $C = (v_1, e_1, \dots, v_\ell, e_\ell, v_1)$ in the hypergraph $H - S'$. This implies that $\{v_1, \dots, v_\ell\} \subseteq A \setminus S'$ and $\{e_1, \dots, e_\ell\} \subseteq B \setminus N_G(S')$. Therefore, $v_1 e_1 \dots v_\ell e_\ell v_1$ is a cycle in $G - N_G[S']$, which is a contradiction to the assumption that S' is a solution to (G, k). ◀

B Safeness Proofs of Reduction Rules in Section 3

▶ **Lemma 18.** *Reduction Rule 3.5 is safe.*

Proof. Observe that, the instance $(G', \mathcal{F}', k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Observe that, if S does not contain v, then S is also a solution to $(G', \mathcal{F}', k)$, as $G' - N_{G'}[S] = (G - N_G[S]) - v$, $(G - N_G[S])$ is acyclic and S is also a hitting set of $\mathcal{F}'$. Next, consider the case when $v \in S$. Let $S' = S \setminus \{v\}$. Since v is an isolated vertex in G, we have that $G - N[S']$ is acyclic. Let $u \in Y$, $u \neq v$, then observe that, $S' \cup \{u\}$ is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS, which does not contain v and hence a solution to $(G, \mathcal{F}', k)$.

In the backward direction, let S' be a solution to $(G', \mathcal{F}', k)$. Suppose that, $G - N[S']$ contains a cycle C. Then, since $G' = G - v$, and $d_G(v) = 0$, C is also a cycle in $G' - N[S']$. Observe that, $\mathcal{F} \setminus \{Y\} = \mathcal{F}' \setminus (Y \setminus \{v\})$. Therefore, S' is also a hitting set of $\mathcal{F}$. This implies that S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. ◀

▶ **Lemma 19.** *Reduction Rule 3.6 is safe.*

Proof. Observe that, the instance $(G', \mathcal{F}, k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a solution to $(G, \mathcal{F}, k)$ of HDBFVS. If S does not contain v, then clearly S is also a solution to $(G', \mathcal{F}, k)$ because $G' - N_{G'}[S] = (G - N_G[S]) - v$ and $(G - N_G[S])$ is acyclic. Suppose that, $v \in S$. Let $\{b\} = N_G(v)$. Since Reduction Rule 3.3 is no longer applicable, we have that $d_G(b) > 1$. Let $u \neq v$ be an arbitrary vertex in $N_G(b)$. Then, $S^\star = (S \setminus \{v\}) \cup \{u\}$ is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS because $N_G(v) \subseteq N_G(u)$ and $d_G(v) = 1$. Then, $S^\star$ is also a solution to $(G' = G - v, \mathcal{F}, k)$ of HDBFVS because $G' - N_{G'}[S^\star] = (G - N_G[S^\star]) - v$ and $(G - N_G[S^\star])$ is acyclic.

In the backward direction, let S' be a solution to $(G', \mathcal{F}, k)$. Suppose that, $G - N[S']$ contains a cycle C. Then, since $G' = G - v$, and $d_G(v) = 1$, C is also a cycle in $G' - N[S']$. This implies that S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. ◀

▶ **Lemma 20.** *Reduction Rule 3.7 is safe.*

Proof. Observe that, the instance $(G', \mathcal{F}', k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose that, $u \notin S$. Since $d_G(u) = 1$, we have that u does not participate in any cycle in G. Therefore, any cycle C in $G' - N[S]$ is also a cycle in $G - N[S]$. This implies that $G' - N[S]$ is acyclic.

Observe that, $\mathcal{F} \setminus \{Y\} = \mathcal{F}' \setminus \{Y \setminus \{u\}\}$. This implies that S is a hitting set of $\mathcal{F}'$. Hence, S is also a solution to $(G', \mathcal{F}', k)$ of HDBFVS. Next, consider that $u \in S$. Since u does not participate in any cycle in G, u is only used to hit cycles containing b (recall that when we delete u, we also delete all its neighbors) and to hit the set Y. Since P is a nice path, any cycle that contains b also contains all the vertices in P and hence contains $N_G(u') = b'$, therefore u' can hit all the cycles containing b. Further, since $u' \in Y$, it holds that u' hits the set Y. This implies that $S^* = (S \setminus \{u\}) \cup \{u'\}$ is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. As argued before, S^* is a solution to $(G', \mathcal{F}', k)$ of HDBFVS.

In the backward direction, let S' be a solution to $(G', \mathcal{F}', k)$ of HDBFVS. Since u does not participate in any cycle, any cycle in $G - N[S']$ is also a cycle in $G' - N[S']$. Hence, $G - N[S']$ is acyclic. Also, since $\mathcal{F} \setminus \{Y\} = \mathcal{F}' \setminus \{Y \setminus \{u\}\}$, we have that S' is a hitting set of $\mathcal{F}$. Hence, S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. ◀

▶ **Lemma 21.** *Reduction Rule 3.8 is safe.*

Proof. We first give a proof for Case 1, followed by a proof of Case 2.

Case 1: $X_{v_{i-1}} \neq X_{v_{i+1}}$ or $X_{v_{i-1}} = X_{v_{i+1}} = \emptyset$. The vertices v_{i-1} and b_{i+1} do not have two common neighbors in G', and hence there is no C_4 in G'. Observe that $G'_{\mathcal{F}'}$ is a subgraph of $G_{\mathcal{F}}$. Further, since $G_{\mathcal{F}}$ does not have C_4, $G'_{\mathcal{F}'}$ does not have C_4. Now we claim that there is no C_4 in $G' \cup G'_{\mathcal{F}'}$. There is no C_4 in G', $G'_{\mathcal{F}'}$, and $G \cup G_{\mathcal{F}}$. Thus, if there is a C_4 in $G' \cup G'_{\mathcal{F}'}$, then there is a set $F \in \mathcal{F}'$ such that $|(N_{G'}(b_{i+1}) \cap F| \geq 2$. Notice that $N_{G'}(b_{i+1}) = \{v_{i-1}, v_{i+1}\}$. Since $(X_{v_{i-1}} \neq X_{v_{i+1}}$ or $X_{v_{i-1}} = X_{v_{i+1}} = \emptyset)$ and $|X_{v_{i-1}}|, |X_{v_{i+1}}| \leq 1$ (because P does not have any special vertex), there is no set $F \in \mathcal{F}'$ such that $\{v_{i-1}, v_{i+1}\} \subseteq F$. Thus, we have proved that there is no C_4 in $G' \cup G'_{\mathcal{F}'}$. This implies that the instance $(G', \mathcal{F}', k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose that, $v_i \notin S$. Then, we claim that S is also a solution to $(G', \mathcal{F}', k)$ of HDBFVS. Suppose not, then either there exists a cycle C in $G' - N_{G'}[S]$ or there exists a set $Z \in \mathcal{F}'$ such that $S \cap Z = \emptyset$. First consider the former case. If C does not contain the edge $v_{i-1}b_{i+1}$, then C is also a cycle in $G - N_G[S]$, which is a contradiction. Therefore, C contains the edge $v_{i-1}b_{i+1}$. But, then we get a cycle in $G - N_G[S]$ by replacing the edge $v_{i-1}b_{i+1}$ in C by the path $v_{i-1}b_iv_ib_{i+1}$. This is a contradiction to the assumption that $(G - N[S])$ is acyclic. Now, consider the later case. Note that S hits $\mathcal{F} \setminus \{Y\}$ and $Y \setminus \{v_i\}$ (since $v_i \notin S$). Thus, it implies that S is a hitting set of $\mathcal{F}'$. Hence, S is also a solution to $(G', \mathcal{F}', k)$ of HDBFVS. Next, consider that $v_i \in S$. Since P is a degree two nice path in G, any cycle that contains a vertex from $N[v_i]$ also contains all the vertices in P. In particular, it contains v_j, and v_j hits all the cycles that any vertex in $N[v_i]$ hits. Also, observe that, $v_j \in Y$ and hence v_j hits the set Y. This implies that $S^\star = S \setminus \{v_i\} \cup \{v_j\}$ is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. As argued before $S^\star$ is a solution to $(G', \mathcal{F}', k)$ of HDBFVS.

In the backward direction, let S' be a solution to $(G', \mathcal{F}', k)$ of HDBFVS. We claim that S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose not. Then, either there exists a cycle C in $G - N_G[S']$ or there exists a set $Z \in \mathcal{F}$ such that $S' \cap Z = \emptyset$. First consider the former case. If C does not contain any edge from the path P, then C is also a cycle in $G' - N_{G'}[S']$, which is a contradiction. Therefore, at least one edge from the path P is part of C. Then, since P is a degree two nice path in G, P is a subpath of C. Then, we get a cycle C' in $G' - N_{G'}[S']$ by replacing the subpath $v_{i-1}b_iv_ib_{i+1}$ in C by $v_{i-1}b_{i+1}$. This is a contradiction to the assumption that S' is a solution to $(G', \mathcal{F}', k)$. Now, consider the later case. Since $\mathcal{F} \setminus \{Y\} = \mathcal{F}' \setminus \{Y \setminus \{v_i\}\}$ and $|Y| \geq 2$, we have that S' is a hitting set of $\mathcal{F}$. Hence, S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS.

Case 2: $X_{v_{i-1}} = X_{v_{i+1}} = \{Y^*\}$. The vertices v_{i-2} and b_{i+1} do not have two common neighbors in G', and hence there is no C_4 in G'. Observe that $G'_{\mathcal{F}'}$ is a subgraph of $G_{\mathcal{F}}$. Further, since $G_{\mathcal{F}}$ does not have C_4, $G'_{\mathcal{F}'}$ does not have C_4. Next we claim that there is no C_4 in $G' \cup G'_{\mathcal{F}'}$. By item (ii) of Observation 3.1, $X_{v_{i-2}} \neq X_{v_{i-1}}$. This implies that $X_{v_{i-2}} \neq X_{v_{i+1}}$. Note that, there is no C_4 in G', $G'_{\mathcal{F}'}$, and $G \cup G_{\mathcal{F}}$. Thus, if there is a C_4 in $G' \cup G'_{\mathcal{F}'}$, then there exists a set $F \in \mathcal{F}'$ such that $|(N_{G'}(b_{i+1}) \cap F| \geq 2$. Notice that $N_{G'}(b_{i+1}) = \{v_{i-2}, v_{i+1}\}$. Since $X_{v_{i-2}} \neq X_{v_{i+1}}$ and $|X_{v_{i-2}}|, |X_{v_{i+1}}| \leq 1$ (because P does not have any special vertex), there is no set $F \in \mathcal{F}'$ such that $\{v_{i-2}, v_{i+1}\} \subseteq F$. Thus, we have proved that there is no C_4 in $G' \cup G'_{\mathcal{F}'}$. This implies that the instance $(G', \mathcal{F}', k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a minimal solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose $v_{i-1} \in S$ or $v_i \in S$. Consider the case $v_{i-1} \in S$. Then, we claim that $S^\star = (S \setminus \{v_{i-1}\}) \cup \{v_{i+1}\}$ is also a solution to $(G, \mathcal{F}, k)$. Since P is a nice path, any cycle that contains a vertex of P must contain all the vertices of P. Thus, all the cycles containing a vertex from $N[v_{i-1}]$, also contain v_{i+1}. Therefore v_{i+1} hits all those cycles that $N[v_{i-1}]$ hits. Since $X_{v_{i-1}} = X_{v_{i+1}} = \{Y^\star\}$, v_{i+1} and v_{i-1} hits the same set (only one) from $\mathcal{F}$. Now suppose that $v_i \in S$. Then, we claim that $S' = (S \setminus \{v_i\}) \cup \{v_j\}$ is a solution to $(G, \mathcal{F}, k)$. Since all the cycles containing a vertex from $N[v_i]$, also contain v_j, therefore v_j hits all the cycles that $N[v_i]$ hits. Since $X_{v_i} = X_{v_j} = \{Y^\star\}$, v_i and v_j hits the same set (only one) from $\mathcal{F}$.

Thus, if $(G, \mathcal{F}, k)$ is a YES-instance, then there is a solution S such that $v_{i-1}, v_i \notin S$. Then, we claim that S is also a solution to $(G', \mathcal{F}', k)$ of HDBFVS. Suppose not, then either there exists a cycle C in $G' - N_{G'}[S]$ or there exists a set $Z \in \mathcal{F}'$ such that $S \cap Z = \emptyset$. First consider the former case. If C does not contain the edge $v_{i-2}b_{i+1}$, then C is also a cycle in $G - N_G[S]$, which is a contradiction. Therefore, C contains the edge $v_{i-2}b_{i+1}$. But, then we get a cycle in $G - N_G[S]$ by replacing the edge $v_{i-2}b_{i+1}$ in C by the path $v_{i-2}b_{i-1}v_{i-1}b_iv_ib_{i+1}$. This is a contradiction to the assumption that $(G - N[S])$ is acyclic. Now, consider the later case. Note that S hits $\mathcal{F} \setminus \{Y, Y^\star\}$ and $\{Y^\star \setminus \{v_{i-1}\}, Y \setminus \{v_i\}\}$ (since $v_{i-1}, v_i \notin S$). Thus, it implies that S is a hitting set of $\mathcal{F}'$. Hence, S is also a solution to $(G', \mathcal{F}', k)$ of HDBFVS.

In the backward direction, let S' be a solution to $(G', \mathcal{F}', k)$ of HDBFVS. We claim that S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose not. Then, either there exists a cycle C in $G - N_G[S']$ or there exists a set $Z \in \mathcal{F}$ such that $S' \cap Z = \emptyset$. First consider the former case. If C does not contain any edge from the path P, then C is also a cycle in $G' - N_{G'}[S']$, which is a contradiction. Therefore, at least one edge from the path P is part of C. Since P is a degree two nice path in G, P is a subpath of C. Thus, we get a cycle C' in $G' - N_{G'}[S']$ by replacing the subpath $v_{i-2}b_{i-1}v_{i-1}b_iv_ib_{i+1}$ in C by $v_{i-2}b_{i+1}$. This is a contradiction to the assumption that S' is a solution to $(G', \mathcal{F}', k)$. Now, consider the later case. Since $\mathcal{F} \setminus \{Y, Y^\star\} = \mathcal{F}' \setminus \{Y^\star \setminus \{v_{i-1}\}, Y \setminus \{v_i\}\}$ and $|Y|, |Y^\star| \geq 2$, we have that S' is a hitting set of $\mathcal{F}$. Hence, S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. ◀

▶ **Lemma 22.** *Reduction Rule 3.9 is safe.*

Proof. Observe that, the instance $(G', \mathcal{F}, k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose that $v_2 \notin S$. Then, we claim that S is also a solution to $(G', \mathcal{F}, k)$ of HDBFVS. Suppose not, then there exists a cycle C in $G' - N_{G'}[S]$. If C does not contain the edge v_1b_3, then C is also a cycle in $G - N_G[S]$, which is a contradiction. Therefore, C contains the edge v_1b_3. But, then we get a cycle in $G - N_G[S]$ by replacing the edge v_1b_3 in C by the path $v_1b_2v_2b_3$. This is a contradiction to the assumption that $(G - N[S])$ is acyclic. Hence, S is also a solution to $(G', \mathcal{F}, k)$ of HDBFVS. Next, consider that $v_2 \in S$. Since P is a degree two nice path, any cycle that contains v_2 also contains all the vertices in P and hence contains v_1.

Therefore $S^\star = S \setminus \{v_2\} \cup \{v_1\}$ is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. As argued before S^* is a solution to $(G', \mathcal{F}, k)$ of HDBFVS.

In the backward direction, let S' be a solution to $(G', \mathcal{F}, k)$ of HDBFVS. We claim that S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. Suppose not. Then, there exists a cycle C in $G - N_G[S']$. If C does not contain any edges from the path P, then C is also a cycle in $G' - N_{G'}[S']$, which is a contradiction. Therefore, at least one edge from the path P is part of C. Since P is a degree two nice path in G, P is a subpath in C. Thus, we get a cycle C' in $G' - N_{G'}[S']$ by replacing the subpath $v_1 b_2 v_2 b_3$ in C by $v_1 b_3$. This is a contradiction to the assumption that S' is a solution to $(G', \mathcal{F}, k)$. Hence, S' is also a solution to $(G, \mathcal{F}, k)$ of HDBFVS. ◀

C Missing proofs from Section 3

C.1 Proof of Lemma 5

Proof. Observe that, $(G = (A \uplus B, E'), \mathcal{F}, k)$ is a valid instance of HDBFVS.

In the forward direction, let S be a solution to (H_1, H_2, k) of HHFVS. We claim that S is also a solution to $(G = (A \uplus B, E'), \mathcal{F}, k)$ of HDBFVS. Suppose not. Then, either there exists a cycle $C = v_1 e_1 \ldots v_\ell e_\ell v_1$ in the graph $G - N_G[S]$ such that for each $i \in [\ell]$, $v_i \in A$, $e_i \in B$ and $v_i e_i \in E'$, and $e_\ell v_1 \in E'$, or S does not hit a set $Y \in \mathcal{F}$. The former case implies that, $e_1, \ldots, e_\ell$ are hyperedges in $H_1 - S$, and $\{v_1, \ldots, v_\ell\} \subseteq V(H_1) \setminus S$. Then, $(v_1, e_1, \ldots, v_\ell, e_\ell, v_1)$ is a cycle in the hypergraph $H_1 - S$. This is a contradiction to the assumption that $H_1 - S$ is acyclic. The later case implies that, there is an edge Y in $H_2 - S$, which is a contradiction to the assumption that $H_2 - S$ is edgeless (that is, S is a hitting set for H_2).

In the backward direction, let S' be a solution to $(G = (A \uplus B, E'), \mathcal{F}, k)$. We claim that S' is also a solution to (H_1, H_2, k) of HDBFVS. Suppose not. Then, either there exists a cycle $C = (v_1, e_1, \ldots, v_\ell, e_\ell, v_1)$ in the hypergraph $H_1 - S'$, or there exists an edge $Y \in H_2 - S$. The former case implies that, $\{v_1, \ldots, v_\ell\} \subseteq A \setminus S'$ and $\{e_1, \ldots, e_\ell\} \subseteq B \setminus N_G(S')$. Therefore, $v_1 e_1 \ldots v_\ell e_\ell v_1$ is a cycle in $G - N_G[S']$, which is a contradiction to the assumption that $G - N[S']$ is acyclic. The later case implies that, S' does not hit the set $Y \in \mathcal{F}$, a contradiction to the assumption that S' is a hitting set for $\mathcal{F}$. ◀

C.2 Proof of Lemma 11

Proof. By Lemma 6, the number of special vertices in P is bounded by $\binom{|\mathcal{F}|}{2}$. Let P' be a maximum length subpath of P such that P' is a nice path. That is, P' does not contain any special vertices. Then, by Lemma 10, the number of vertices in P' that are adjacent to a vertex in $V_{=1}(G)$ in G is bounded by $\binom{|\mathcal{F}|}{2} + |\mathcal{F}|$. Let P'' be a maximum length subpath of P' such that P'' does not contain any vertex that is adjacent to a vertex in $V_{=1}(G)$ in G. Then, by Observation 3.2, either the length of P'' is bounded by 10, or the number of easy vertices in P'' is bounded by $|\mathcal{F}| + 2$. Let $P^\star$ be a maximum length subpath of P'' such that $P^\star$ does not contain any easy vertices. Then, since Reduction Rule 3.9 is no longer applicable, the length of $P^\star$ is bounded by 7. Therefore, we have that the length of P'' is bounded by $7(|\mathcal{F}| + 3)$. This implies that the length of P' is bounded by $7(|\mathcal{F}| + 3)(\binom{|\mathcal{F}|}{2} + |\mathcal{F}| + 1) \leq (35|\mathcal{F}|^3 + 21)$. Hence, the length of P is bounded by $(35|\mathcal{F}|^3 + 21)(\binom{\mathcal{F}}{2} + 1) \leq 63|\mathcal{F}|^5 + 21$. ◀

D Feedback Vertex Sets on General Hypergraphs: Proof of Theorem 1

In order to prove Theorem 1 we give a polynomial time parameter preserving reduction from Set Cover to HFVS. In Set Cover (SC), we are given a universe U, a family $\mathcal{F}$ of sets over U, and a positive integer k, and the question is whether there is a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k, such that $\bigcup_{F \in \mathcal{F}'} F = U$. It is well known that Set Cover is W[2]-hard [10, Theorem 13.28].

Given an instance $(U, \mathcal{F}, k)$ of SC, we construct an instance (H, k) of HFVS as follows. For each element $u \in U$, let X_u be the family of sets in $\mathcal{F}$ that contain u. For each $F \in \mathcal{F}$, we add a vertex w_F in H. Furthermore, for each $u \in U$, we add $2(k+1)$ vertices $\{u_1, u'_1, \ldots, u_{k+1}, u'_{k+1}\}$ in H. Hence, $V(H) = \{w_F \mid F \in \mathcal{F}\} \cup \{u_1, u'_1, \ldots, u_{k+1}, u'_{k+1} \mid u \in U\}$. Now, we explain the construction of hyperedges of H. For each $u \in U$, we introduce a hyperedge $e_u = \{w_F \mid F \in X_u\}$ containing vertices corresponding to the sets in X_u. Also, for each $u \in U$, we add hyperedges $e_u \cup \{u_i\}$, $\{u_i, u'_i\}$, $e_u \cup \{u'_i\}$, for all $i \in [k+1]$. This completes the construction. Towards the proof of Theorem 1, we give the following lemma.

▶ **Lemma 23.** *$(U, \mathcal{F}, k)$ is a* YES*-instance of* SC *if and only if (H, k) is a* YES*-instance of* HFVS.

Proof. In the forward direction, let $\mathcal{S}$ be a solution to $(U, \mathcal{F}, k)$ of SC. We claim that $Z = \{w_F \mid F \in \mathcal{S}\}$ is a feedback vertex set of size at most k in H. Since $|\mathcal{S}| \leq k$, we have that $|Z| \leq k$. Next, we prove that Z is a feedback vertex set in H. Since $\mathcal{S}$ is a set cover, the only hyperedges of H present in $H - Z$ are $\{\{u_i, u'_i\} \mid u \in U, i \in [k+1]\}$. Notice that $\{\{u_i, u'_i\} \mid u \in U, i \in [k+1]\}$ are pairwise disjoint. This implies that $H - Z$ is acyclic.

In the reverse direction, let Z be a solution to (H, k) of HFVS. Let $Z' = Z \setminus \{u_1, u'_1, \ldots, u_{k+1}, u'_{k+1} \mid u \in U\}$. That is, Z' contains only those vertices of Z that correspond to some set in $\mathcal{F}$. Let $\mathcal{S} = \{F \mid w_F \in Z'\}$. Since $|Z'| \leq |Z| \leq k$, we have that $|\mathcal{S}| \leq k$. Next we claim that $\mathcal{S}$ is a set cover of $(U, \mathcal{F}, k)$. Towards that, we choose an arbitrary element $u \in U$ and prove that there is a set $F \in \mathcal{S}$ which contains u. Let J be an arbitrary set in $\mathcal{F}$ such that $u \in J$. Notice that there are $k+1$ triangles $(w_J, e_u \cup \{u_i\}, u_i, \{u_i, u'_i\}, u'_i, e_u \cup \{u'_i\}, w_J), 1 \leq i \leq k+1$, in H. This implies that at least one vertex w_F in e_u must belong to the feedback vertex set Z (and Z'). However, u belongs to F and F is in $\mathcal{S}$. Hence u is covered by $\mathcal{S}$. This completes the proof. ◀

E Feedback Vertex Sets on d-Hypergraphs: Proof of Theorem 2

In this section we design an FPT algorithm for HFVS on d-hypergraphs. Towards this, we will prove the following result about DFVSB, from which Theorem 2 will follow as a corollary.

▶ **Theorem 24.** *There is a deterministic algorithm for* DFVSB *running in time $\mathcal{O}(2^{7k} d^{2k+1} n(n+m) + n^2(n+m))$, where the input is a bipartite graph G with bipartition $V(G) = A \uplus B$, and $d = \max_{b \in B} d_G(b)$.*

Towards designing an FPT algorithm for DFVSB, we use the well-known iterative compression technique [10, Chapter 4]. Usually, the primary step in the technique of iterative compression involves solving a "disjoint compression version" of the problem. In our case, the disjoint compression version of the problem is defined as follows.

> d-Disjoint Dominating Bounded Bipartite FVS (d-DDBB-FVS)
> **Input:** A d-bipartite graph $G = (A \uplus B, E)$, a positive integer k, and a vertex subset $W \subseteq A$ such that $G - N[W]$ is acyclic.
> **Question:** Is there a set $S \subseteq A \setminus W$ of at most k vertices such that $G - N[S]$ is acyclic?

We denote an instance of d-DDBB-FVS as (G, k, W), where G is the input graph with bipartition $A \uplus B$, k is the parameter (the solution size), and W is a set such that for a solution S, it holds that $S \subseteq A \setminus W$. The main result of the section is the following lemma.

▶ **Lemma 25.** *Given an instance $((A \uplus B, E), k, W)$ of d-DDBB-FVS, there exists an algorithm that gives a solution in time $\mathcal{O}((8d)^{k+\gamma(G_W)}(n + m) + n(n + m))$, where $d = \max_{w \in B} d(w)$, $n = |V(G)|$, $m = |E(G)|$, and $\gamma(G_W)$ is the number of connected components in the subgraph $G_W = G[W \cup \{b \in B : N(b) \subseteq W\}]$.*

Assuming Lemma 25 one can prove Theorem 24, somewhat similar to the way it is done for FVS on graphs (see Section 4.1 in [10]). We use the following observation in the proof of Theorem 24.

▷ Observation E.1. Let $(G = (A \uplus B, E), k)$ be an instance of DFVSB, and $B' \subseteq B$. If $(G' = (A \uplus B', E(A, B')), k)$ is a NO-instance of DFVSB, then $(G = (A \uplus B, E), k)$ is a NO-instance of DFVSB.

Proof. Any solution to $(G = (A \uplus B, E), k)$ is also a solution to $((A \uplus B', E(A, B')), k)$. ◀

Now we give a proof sketch of Theorem 24 assuming Lemma 25.

Proof sketch of Theorem 24. We employ the method of iterative compression to prove Theorem 24. Towards that, we iteratively apply Lemma 25. Let $(G = (A \uplus B, E), k)$ be the input of DFVSB. Let $B = \{b_1, \ldots, b_r\}$. If $r \leq k + 1$, then any subset $A' \subseteq A$ of size at most $r - 1$ that contains a neighbor of b_i for all $i \in [r - 1]$ is a solution to (G, k). That is, if $r \leq k + 1$, then (G, k) is a YES-instance. Otherwise, we proceed as follows.

Initially we consider the instance $J_1 = (G_1 = (A \uplus B_1), k)$ of DFVSB, where $B_1 = \{b_1, \ldots, b_{k+2}\}$. Let $W_1 = \{v_1, \ldots, v_{k+1}\}$ be an arbitrary subset of A such that $N(b_j) \cap W_1 \neq \emptyset$ for all $j \in [k + 1]$. Clearly, W_1 is a dominating feedback vertex set of size $k + 1$ for G_1. To compute a dominating feedback vertex set of size at most k, for each subset $S \subseteq W_1$ of size at most k (a potential guess of the intersection of a hypothetical solution with W_1), we use Lemma 25 to check whether there exists a solution to the instance $(G_1' = G_1 - N[S], k - |S|, W_1 \setminus S)$ of d-DDBB-FVS. If no such solution exists for any choice of the subset of W_1, then clearly J_1 is a NO-instance of DFVSB due to observation E.1.

Otherwise, if there is a subset $S_1 \subseteq W_1$ of size at most k, such that Q_1 is a solution for $(((A \setminus S_1) \uplus (B_1 \setminus N(S_1)), E(A \setminus S_1, B_1 \setminus N(S_1))), k - |S_1|, W_1 \setminus S_1)$ of d-DDBB-FVS, then $S_1 \cup Q_1$ is a solution of size at most k for the instance J_1. Next, we construct an instance $J_2 = (G_2 = (A \uplus B_2, E(A, B_2)), k)$ of DFVSB, where $B_2 = \{b_1, \ldots, b_{k+3}\}$. Let $W_2 = S_1 \cup Q_1 \cup \{v\}$, where v is an arbitrary vertex in $N(b_{k+3})$. Notice that $G_2 - N[W_2]$ is a subgraph of $G_1 - N[S_1 \cup Q_1]$ which is a forest. That is, W_2 is a dominating feedback vertex set of G_2 of size at most $k + 1$. Now we repeat the same process as described above to "compress" the solution size of J_2 to at most k. At each iteration, if there exists a solution W_i of size at most k for the instance J_i, then in step $i + 1$, $W_i \cup \{v\}$ is a dominating feedback vertex set for $G_{i+1} = (A \uplus B_{i+1}, E(A, B_{i+1}))$, where $B_{i+1} = B_i \cup \{b_{k+2+i}\}$ and $v \in N(b_{k+2+i})$, and we continue the same process.

Finally, notice that $J_{r-(k+1)}$ is actually the input instance (G,k), and we get a solution to $J_{r-(k+1)}$ at the end of the algorithm (if (G,k) is a YES-instance). More formally, at step $i \in [r-(k+1)]$, we have an instance $J_i = (G_i = (A \uplus B_i, E(A,B_i)), k)$, where $B_i = \{b_1, \ldots, b_{k+1+i}\}$, and a dominating feedback vertex set W_i' of G_i of size at most $k+1$. Then, by applying Lemma 25 at most 2^{k+1} times we obtain a solution W_i of size at most k for the instance J_i (if it exists). If there does not exist a solution for J_i, then (G,k) is a NO-instance.

Since we apply Lemma 25 at most $2^{k+1}|B|-(k+1)$ times and the number of connected components of G_W in each application of Lemma 25 is at most $k+1$, the total running time is upper bounded by $\mathcal{O}(2^k(8d)^{2k+1}n(n+m)+n^2(n+m)) = \mathcal{O}(2^{7k}d^{2k+1}n(n+m)+n^2(n+m))$, where $n=|V(G)|$ and $m=|E(G)|$. ◀

The rest of the section is devoted to the proof of Lemma 25. Towards proving Lemma 25, we design a branching algorithm consisting of three branching rules and some simple reduction rules. To bound the running time, we define a *measure* associated with an instance of d-DDBB-FVS, and this measure decreases by at least one during each application of the branching rules. It does not increase during the application of any of the reduction rules. Moreover, the number of children for each node in the branching tree is bounded by $\mathcal{O}(d)$. For an instance $(G=(A \uplus B, E), k, W)$ of d-DDBB-FVS, recall that $G_W = G[W \cup \{b \in B \colon N_G(b) \subseteq W\}]$, and $\gamma(G_W)$ is the number of connected components in G_W. We define the measure associated with the instance (G,k,W) of d-DDBB-FVS as,

$$\mu(G,k,W) = k + \gamma(G_W)$$

For a reduction rule that takes an instance (G,k,W) of d-DDBB-FVS and outputs another instance (G',k',W') of d-DDBB-FVS, we say that the reduction rule is safe if the following holds: (i) (G,k,W) is a YES-instance if and only if (G',k',W') is a YES-instance, and (ii) $\mu(G',k',W') \leq \mu(G,k,W)$. A branching rule for d-DDBB-FVS, takes an instance (G,k,W) and outputs a collection of instances $(G_1,k_1,W_1),\ldots,(G_\ell,k_\ell,W_\ell)$. We say that the branching rule is safe if the following holds: (i) (G,k,W) is a YES-instance if and only if (G_i,k_i,W_i) is a YES-instance for some $i \in [\ell]$, and (ii) for each $i \in [\ell]$, $\mu(G_i,k_i,W_i) < \mu(G,k,W)$.

▷ Reduction Rule E.1. Let (G,k,W) be an instance of d-DDBB-FVS. If $k=0$ and G is not acyclic, then return that (G,k,W) is a NO-instance of d-DDBB-FVS.

▷ Reduction Rule E.2. Let (G,k,W) be an instance of d-DDBB-FVS. If G is acyclic and $k \geq 0$, then return $\emptyset$ and STOP.

The correctness of the above reduction rules follows from the fact that (G,k,W) is a YES-instance of d-DDBB-FVS and $\emptyset$ is a solution to (G,k,W).

▷ Reduction Rule E.3. Let (G,k,W) be an instance of d-DDBB-FVS. Let $v \in V(G)$ be a vertex of degree 0 in G. Then, output $(G-v,k,W \setminus \{v\})$.

It is easy to see that the above reduction rules are safe and can be applied in polynomial time.

▷ Reduction Rule E.4. Let $(G=(A \uplus B,E),k,W)$ be an instance of d-DDBB-FVS and $b \in B$ be a vertex of degree 1 in G. Then, output $(G-b,k,W)$.

▶ **Lemma 26.** *Reduction Rule E.4 is safe.*

Proof. Since $d_G(b) = 1$, there is no cycle in G containing b. Therefore, any solution to $(G - b, k, W)$ is also a solution to (G, k, W) and vice versa. Let $G' = G - b$. Since $d_G(b) \leq 1$, $\gamma(G'_W) \leq \gamma(G_W)$. Therefore, $\mu(G', k, W) \leq \mu(G, k, W)$ and Reduction Rule E.4 is safe. ◀

▷ Reduction Rule E.5. Let $(G = (A \uplus B, E), k, W)$ be an instance of d-DDBB-FVS and $v \in A \setminus W$ be a vertex of degree 1 in G. Let $N_G(v) = \{b\}$. Moreover, either $N_G(b) \setminus (W \cup \{v\}) \neq \emptyset$ or $d_G(b) = 2$. Then, output $(G - v, k, W)$.

▶ **Lemma 27.** *Reduction Rule E.5 is safe.*

Proof. First consider the case $N_G(b) \setminus (W \cup \{v\}) \neq \emptyset$. Since $d_G(v) = 1$, any solution to $(G - v, k, W)$ is also a solution to (G, k, W). Now suppose that, (G, k, W) is a YES-instance. Let u be an arbitrary vertex in $N_G(b) \setminus (W \cup \{v\})$ and $G' = G - v$. First we claim that there is a solution S to (G, k, W) that does not contain v. If there exists a solution S' to (G, k, W) that contains v, then $S^\star = (S' \setminus \{v\}) \cup \{u\}$ is a solution to (G, k, W), because $N_G(v) \subseteq N_G(u)$ and $d_G(v) = 1$. Let S be a solution to (G, k, W) such that $v \notin S$. Then, S is also a solution to $(G' = G - v, k, W)$ because $G' - N_{G'}[S] = (G - N_G[S]) - v$, and $(G - N_G[S])$ is acyclic. Notice that $G'_W = G_W$. Therefore, $\mu(G', k, W) \leq \mu(G, k, W)$.

Next, we consider the case $d_G(b) = 2$. Here, there is no cycle in G that contains either b or v. This implies that, if S is a solution to (G, k, W), then $S \setminus \{v\}$ is a solution to $(G - v, k, W)$. Since $d_G(v) = 1$, any solution to $(G' = G - v, k, W)$ is also a solution to (G, k, W). Also, since $G'_W = G_W$, we have that $\mu(G', k, W) \leq \mu(G, k, W)$. ◀

▷ Reduction Rule E.6. Let (G, k, W) be an instance of d-DDBB-FVS. Let $b_1 v_1 b_2 v_2 b_3 v_3 b_4$ be a path in G such that $v_1 b_2 v_2 b_3 v_3$ is a degree two path in G, $\{b_1, \ldots, b_4\} \subseteq B$ and $\{v_1, v_2, v_3\} \subseteq A \setminus W$. Now, let G' be the graph obtained by deleting the vertices b_2, v_2 from G and adding a new edge $v_1 b_3$, i.e. $G' = (G - \{v_2, b_2\}) + v_1 b_3$. Then, output (G', k, W).

▶ **Lemma 28.** *Reduction Rule E.6 is safe.*

Proof. First, we prove that (G, k, W) is a YES-instance of d-DDBB-FVS if and only if (G', k, W) is a YES-instance of d-DDBB-FVS. In the forward direction, let S be a solution to (G, k, W) of d-DDBB-FVS. Suppose that, $v_2 \notin S$. Then, we claim that S is also a solution of (G', k, W). Suppose not, then there exists a cycle C in $G' - N_{G'}[S]$. If C does not contain the edge $v_1 b_3$, then C is also a cycle in $G - N_G[S]$, which is a contradiction. Therefore, C contains the edge $v_1 b_3$. But, then we get a cycle in $G - N_G[S]$ by replacing the edge $v_1 b_3$ in C by the path $v_1 b_2 v_2 b_3$. This is a contradiction to the assumption that S is a solution to (G, W, k). Now, consider the case $v_2 \in S$. Then, $S' = (S \setminus \{v_2\}) \cup \{v_1\}$ is a solution to (G', k, W) because $S' \cap W = \emptyset$ and any cycle in G which contains any of the vertices in $\{b_2, v_2, b_3\}$ also contains v_1.

For the backward direction, let $S^\star$ be a solution to (G', k, W) of d-DDBB-FVS. Clearly, $S^\star \subseteq A \setminus W$. We claim that $S^\star$ is also a solution to (G, k, W). Suppose not. Then, there exists a cycle C in $G - N_G[S^\star]$. If C does not contain any edges from $\{v_1 b_2, b_2 v_2, v_2 b_3\}$, then C is also a cycle in $G' - N_{G'}[S^\star]$, which is a contradiction. Therefore, at least one edge from $\{v_1 b_2, b_2 v_2, v_2 b_3\}$ is part of C. Then, since $v_1 b_2 v_2 b_3 v_3$ is a degree two path in G, $b_1 v_1 b_2 v_2 b_3 v_3 b_4$ is a subpath in C. Then, we get a cycle C' in $G' - N_{G'}[S^\star]$ by replacing the subpath $v_1 b_2 v_2 b_3$ in C by $v_1 b_3$. This is a contradiction to the assumption that $S^\star$ is a solution to (G', k, W). Hence, $S^\star$ is also a solution to (G, k, W).

Next, we prove that $\mu(G', k, W') \leq \mu(G, k, W)$. Since $v_1, v_2, v_3 \notin W$, we have that $b_1, b_2, b_3, b_4 \notin V(G_W)$. Therefore, we have that $G_W = G'_{W'}$ and hence, $\mu(G', k, W') = \mu(G, k, W)$. This completes the proof of the lemma. ◀

▷ Branching Rule 1. Let (G, k, W) be an instance of d-DDBB-FVS and let $b \in B$ be a vertex such that $N_G(b) \setminus W \neq \emptyset$ and $|N_G(b) \cap W| \geq 2$. Let $z, z' \in N_G(b) \cap W$ be two distinct vertices and $N_G(b) \setminus W = \{u_1, \ldots, u_\ell\}$. If z and z' are in the same connected component of G_W, then we branch into the following instances: $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$. If z and z' are in two distinct connected components of G_W, then we branch into the following instances: $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$, and $(G, k, W \cup \{u_1, \ldots, u_\ell\})$.

▶ **Lemma 29.** *Branching Rule 1 is safe.*

Proof. First consider the case that z and z' are in the same connected component of G_W. If (G, k, W) is a NO-instance, then clearly all the instances $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$ are NO-instances. Since z and z' are in the same connected component of G_W, there is a cycle C in $G[V(G_W) \cup \{b\}]$. Also, notice that $N_G(V(C) \cap B) \setminus W \subseteq \{u_1, \ldots, u_\ell\}$. That is, if (G, k, W) is a YES-instance, then any solution will contain a vertex from $\{u_1, \ldots, u_\ell\}$. Therefore, if (G, k, W) is a YES-instance, then at least one of the instances $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$ is a YES-instance. Now we prove that $\mu(G - N[u_i], k-1, W) \leq \mu(G, k, W) - 1$, for all $i \in [\ell]$. Towards that, we fix an arbitrary $i \in [\ell]$. Let $G' = G - N[u_i]$. Since $u_i \in A \setminus W$, $G_W = G'_W$. This implies that, $\gamma(G'_W) = \gamma(G_W)$. Therefore, $\mu(G', k-1, W) = k - 1 + \gamma(G'_W) = k + \gamma(G_W) - 1 = \mu(G, k, W) - 1$.

Next, consider the case that z and z' are in two different connected components of G_W. If (G, k, W) is a NO-instance, then clearly all the instances $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$, and $(G, k, W \cup \{u_1, \ldots, u_\ell\})$ are NO-instances. Suppose that, (G, k, W) is YES-instance. Let S be a solution to (G, k, W). If $S \cap \{u_1, \ldots, u_\ell\} \neq \emptyset$, then at least one of $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$ is a YES-instance. Otherwise, S is a solution to $(G, k, W \cup \{u_1, \ldots, u_\ell\})$. The proof of $\mu(G - N[u_i], k-1, W) \leq \mu(G, k, W) - 1$ for all $i \in [\ell]$, given in the above paragraph holds in this case as well. Finally, we prove that $\mu(G, k, W \cup \{u_1, \ldots, u_\ell\}) \leq \mu(G, k, W) - 1$. Note that, it is enough to prove that $\gamma(G_{W'}) \leq \gamma(G_W) - 1$, where $W' = W \cup \{u_1, \ldots, u_\ell\}$. Observe that, each connected component in $G_{W'}$ contains a vertex from W', as Reduction Rule E.3 is no longer applicable. Moreover, G_W is a subgraph of $G_{W'}$ and there is a connected component in $G_{W'}$ containing z and z', because $z, z' \in N_G(b)$ and $b \in V(G_{W'})$. Also, notice that in this case z and z' belong to different connected components in G_W. This implies that, $\gamma(G_{W'}) \leq \gamma(G_W) - 1$. This completes the proof of the lemma. ◀

▷ Branching Rule 2. Let (G, k, W) be an instance of d-DDBB-FVS. If there exists a path/cycle $P = b_0 v_0 \ldots b_r v_r b_{r+1}$ in G, such that $\{v_0, \ldots, v_r\} \subseteq A \setminus W$, $0 \leq r \leq 6$, and there is a cycle in the graph $G[V(G_W) \cup V(P)]$, then we branch into the following instances: $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$, where $\{u_1, \ldots, u_\ell\} = N_G(\{b_0, \ldots, b_{r+1}\}) \setminus W$.

▶ **Lemma 30.** *Branching Rule 2 is safe.*

Proof. If (G, k, W) is a NO-instance, then clearly all the instances $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$ are NO-instances. Now, we prove that if (G, k, W) is a YES-instance, then at least one of the instances $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$ is a YES-instance. Notice that there exists a cycle C in $G[V(G_W) \cup V(P)]$. Therefore, any solution to (G, k, W) contains a vertex from $N_G(V(C) \cap B) \setminus W$. Since $N_G(b) \subseteq W$ for all $b \in B \cap V(G_W)$, we have that $N_G(V(C) \cap B) \setminus W \subseteq N(\{b_0, \ldots, b_{r+1}\}) \setminus W = \{u_1, \ldots, u_\ell\}$. Therefore, if (G, k, W) is a YES-instance, then at least one of the instances $(G - N[u_1], k-1, W), \ldots, (G - N[u_\ell], k-1, W)$ is a YES-instance as well.

Next, we prove that $\mu(G - N[u_i], k-1, W) = \mu(G, k, W) - 1$ for all $i \in [\ell]$. Towards that, we fix an arbitrary $i \in [\ell]$. Let $G' = G - N[u_i]$. Since $u_i \in A \setminus W$, we have that

$G_W = G'_W$. Therefore, $\mu(G', k-1, W) = k-1+\gamma(G'_W) = k+\gamma(G_W)-1 = \mu(G,k,W)-1$. This completes the proof of the lemma. ◀

▷ Branching Rule 3. Let (G,k,W) be an instance of d-DDBB-FVS. Let $P = b_0v_0,\ldots,b_rv_rb_{r+1}$ be a path in G, such that $0 \le r \le 6$ and $\{v_0,\ldots,v_r\} \subseteq A \setminus W$. Let z and z' be two vertices in two distinct connected components of G_W. If there is path from z to z' in the graph $G[V(G_W) \cup V(P)]$, then we branch into the following instances: $(G-N[u_1], k-1, W), \ldots, (G-N[u_\ell], k-1, W)$, and $(G, k, W \cup \{u_1,\ldots,u_\ell\})$, where $\{u_1,\ldots,u_\ell\} = N_G(\{b_0,\ldots,b_{r+1}\}) \setminus W$.

▶ **Lemma 31.** *Branching Rule 3 is safe.*

Proof. If (G,k,W) is a NO-instance, then clearly all the instances $(G-N[u_1], k-1, W), \ldots, (G-N[u_\ell], k-1, W)$ and $(G, k, W \cup \{u_1,\ldots,u_\ell\})$ are NO-instances. Now we prove that if (G,k,W) is a YES-instance, then at least one of the instances $(G-N[u_1], k-1, W), \ldots, (G-N[u_\ell], k-1, W)$ and $(G, k, W \cup \{u_1,\ldots,u_\ell\})$ is a YES-instance. Let S be a solution to (G,k,W). If $S \cap \{u_1,\ldots,u_\ell\} \neq \emptyset$, then at least one of $(G-N[u_1], k-1, W), \ldots, (G-N[u_\ell], k-1, W)$ is a YES-instance. Otherwise S is a solution to $(G, k, W \cup \{u_1,\ldots,u_\ell\})$.

Next, we prove that $\mu(G-N[u_i], k-1, W) \le \mu(G,k,W)-1$, for all $i \in [\ell]$. Here, the proof follows the arguments similar to those in the proof of Lemma 30. Now we prove that $\mu(G, k, W \cup \{u_1,\ldots,u_\ell\}) \le \mu(G,k,W)-1$. Towards that, it is enough to prove that $\gamma(G_{W'}) \le \gamma(G_W)-1$, where $W' = W \cup \{u_1,\ldots,u_\ell\}$. Notice that each connected component in $G_{W'}$ contains a vertex from W'. Moreover, G_W is a subgraph of $G_{W'}$ and there is a connected component in $G_{W'}$ containing z and z', because $V(P) \subseteq V(G_{W'})$. Also, notice that by our assumption z and z' belong to different connected components in G_W. This implies that, $\gamma(G_{W'}) \le \gamma(G_W)-1$. This completes the proof of the lemma. ◀

Now we are ready to complete the proof of Lemma 25.

Proof of Lemma 25. We design a branching algorithm for the problem. Let (G,k,W) be an instance of d-DDBB-FVS. We prove that we can always apply either one of the reduction rules or one of the branching rules until we reach a solution or a NO-instance. First we test if any of the Reduction Rules E.1, E.2, E.3, E.4, and E.5 is applicable. This can easily be tested in linear time. If any of these reduction rules are applicable, we apply them. Next, we test whether Reduction Rule E.6 is applicable. Towards that, let H be a graph obtained from G by deleting all the vertices in W and the vertices of degree at least 3 in G. Then, for any maximal path P such that the internal vertices of P have degree exactly two in G and $V(P) \cap W = \emptyset$, there exists a component in H which is an induced path containing all the vertices of P. Thus, we can identify such a path $P = b_1v_1b_2v_2b_3v_3b_4$ in G such that the internal vertices of P are degree exactly two in G and $V(P) \cap W = \emptyset$ (if it exists) in linear time. If such a path exists, then we apply Reduction Rule E.6. Next, if Branching Rule 1 is applicable, then we apply it. This can be done in linear time as well.

For rest of the proof, we assume that Reduction Rules E.1–E.6, and Branching Rule 1 are not applicable on (G,k,W). We know that $F = G - N_G[W]$ is acyclic. Since $d_G(b) \ge 2$ for all $b \in B$ (because Reduction Rules E.3 and E.4 are not applicable) and $F = G - N_G[W]$, (i) any vertex $u \in V(F)$ with degree at most 1 in F (i.e., $d_F(u) \le 1$) belongs to $A \setminus W$. Now we claim that (ii) there is no vertex of degree zero in F. Suppose not. Let $v \in V(F)$ be such that $d_F(v) = 0$. Because of statement (i), we have that $v \in A \setminus W$. Since Reduction Rule E.3 is not applicable, we have that $d_G(v) \ge 1$. If $d_G(v) = 1$, then $N_G(b) \setminus (W \cup \{v\}) = \emptyset$ and $d_G(b) > 2$, where $\{b\} = N_G(v)$, as Reduction Rules E.4 and E.5 are not applicable. This implies that, $N_G(b) \setminus W \neq \emptyset$ and $|N_G(b) \cap W| \geq 2$. As a result Branching Rule 1 is applicable, which is a contradiction. Thus, we have proven statement (ii).

Next we prove that (iii) for each $v \in V(F)$ such that degree of v is 1 in F, there is a vertex $b \in N_G(W)$ such that $vb \in E(G)$. Towards that, it is enough to prove that for each $v \in V(F)$ of degree 1 in F, $d_G(v) \geq 2$. If $d_G(v) = 1$, then $N_G(b) \setminus (W \cup \{v\}) = \emptyset$ and $d_G(b) > 2$, where $\{b\} = N_G(v)$, as Reduction Rules E.4 and E.5 are not applicable. This implies that, $N_G(b) \setminus W \neq \emptyset$ and $|N_G(b) \cap W| \geq 2$. As a result Branching Rule 1 is applicable, which is a contradiction. Thus, we have proven statement (iii).

Let Q be a path in F (of length more than 0) such that the end-vertices of Q have degree 1 in F, and all but at most one internal vertex of Q has degree exactly 2 in F. Any forest F containing at least one edge contains such a path and it can be computed in linear time. Since the end-vertices of Q have degree 1 in F, by statement (i), the end-vertices of Q belong to $A \setminus W$. Let $Q = v_0 b_1 \dots b_\ell v_\ell$ for some $\ell \in \mathbb{N}$, where $\{v_1, \dots, v_\ell\} \subseteq A \setminus W$ and $\{b_1, \dots, b_\ell\} \subseteq B \setminus N_G(W)$. Due to statement (iii), there exist vertices $b, b' \in N_G(W)$ (not necessarily distinct), such that $bv_0, b'v_\ell \in E(G)$.

Case 1: $\ell \leq 6$. Let P be the path/cycle $bv_0 b_1 \dots b_\ell v_\ell b'$. Note that, P is a cycle if $b = b'$ and P is a path if $b \neq b'$. If P is a cycle, then Branching Rule 2 is applicable and we apply it. Suppose that, $b' \neq b$. Notice that $b, b' \in N_G(W)$. This implies that, there exist vertices z and z' in W, such that $bz, b'z' \in E(G)$. If z and z' belong to the same connected component in G_W, then either Branching Rule 2 is applicable, or Branching Rule 3 will be applicable due to existence of path P. We apply the branching rule accordingly.

Case 2: $\ell \geq 7$. Recall that, all but at most one vertex in $Q = v_0 b_1 \dots b_\ell v_\ell$ has degree at most 2 in F. If all the vertices in Q have degree at most two in F, then either no vertex v_i, $i \in \{1, \dots, 3\}$ has a neighbor in $N(W)$ and Reduction Rule E.6 is applicable, or there exists a vertex v_i, $i \in \{1, \dots, 3\}$, such that v_i has a neighbor in $N(W)$ and either Branching Rule 2, or Branching Rule 3 is applicable. Next, consider that there exists a vertex in Q with degree more than 2 in F. (a) A vertex in $\{v_1, v_2, v_3, b_1, b_2, b_3\}$ has degree more than 2 in F. (b) A vertex in $\{v_4, v_5, v_6, b_4, b_5, b_6, b_7\}$ has degree more than 2 in F. Without loss of generality let us assume (b) (Other case can be argued similarly). That is, each vertex in $\{v_1, v_2, v_3, b_1, b_2, b_3, \}$ has degree at most 2 in F. First, we prove that there exists $i \in \{1, \dots, 3\}$ such that $N_G(v_i) \cap N_G(W) \neq \emptyset$. Otherwise $v_1 b_2 v_2 b_3 v_3$ is a degree two path in G, and hence, Reduction Rule E.6 is applicable, a contradiction to the assumption that none of the reduction rules are applicable.

Now, we fix $i \in \{1, \dots, 3\}$ such that $N_G(v_i) \cap N_G(W) \neq \emptyset$. Let $b^\star \in N_G(W)$ be such that $v_i b^\star \in E(G)$. Let $Q^\star$ be the subpath of Q between v_0 and v_i and $P^\star$ be the path $bQ^\star b^\star$. Clearly, due to existence of path $P^\star$, either Branching Rule 2 or Branching Rule 3 is applicable. We apply the branching rule accordingly.

Now we do the running time analysis. Let $n = |V(G)|$ and $m = |E(G)|$. Each application of a reduction rule takes linear time. Moreover, after each application of a reduction rule, the number of vertices in the graph drops by at least one. Therefore, the total time taken to apply all the reduction rules together in one branch of the branching tree is upper bounded by $\mathcal{O}(n(n+m))$. Each application of a branching rule takes linear time. The number of branches created during an application of Branching Rules 2 or 3 is at most $8d$. Moreover, after each application of Branching Rules 2 and 3, the measure associated with the instance drops by at least one. Therefore, the total running time is upper bounded by $\mathcal{O}((8d)^{k+\gamma(G_W)}(n+m) + n(n+m))$. This concludes the proof. ◀