Discrete Mathematics and Theoretical Computer Science ....
.... **vol. 26:3 #4 (2024)**

Structural Parameterizations of the Biclique-Free Vertex Deletion Problem

Lito Goldmann[1] Leon Kellerhals[1] Tomohiro Koana[2*]

[1] *Technische Universität Berlin, Germany*
[2] *Utrecht University, The Netherlands*

revisions 7th Feb. 2024, 16th July 2024; accepted 30th Aug. 2024.

In this work, we study the Biclique-Free Vertex Deletion problem: Given a graph G and integers k and $i \leq j$, find a set of at most k vertices that intersects every (not necessarily induced) biclique $K_{i,j}$ in G. This is a natural generalization of the Bounded-Degree Deletion problem, wherein one asks whether there is a set of at most k vertices whose deletion results in a graph of a given maximum degree r. The two problems coincide when $i = 1$ and $j = r+1$. We show that Biclique-Free Vertex Deletion is fixed-parameter tractable with respect to $k + d$ for the degeneracy d by developing a $2^{\mathcal{O}(dk^2)} \cdot n^{\mathcal{O}(1)}$-time algorithm. We also show that it can be solved in $2^{\mathcal{O}(fk)} \cdot n^{\mathcal{O}(1)}$ time for the feedback vertex number f when $i \geq 2$. In contrast, we find that it is W[1]-hard for the treedepth for any integer $i \geq 1$. Finally, we show that Biclique-Free Vertex Deletion has a polynomial kernel for every $i \geq 1$ when parameterized by the feedback edge number. Previously, for this parameter, its fixed-parameter tractability for $i = 1$ was known (Betzler et al., 2012) but the existence of polynomial kernel was open.

Keywords: Fixed-parameter tractability, Kernelization, Structural graph parameterizations, Biclique-free graphs

1 Introduction

The $\mathcal{G}$-Vertex Deletion problem, which, for a graph class $\mathcal{G}$, asks whether a given graph G can be turned into a graph $G' \in \mathcal{G}$ by deleting at most k vertices, is arguably one of the most pervasive and general graph theoretical problems. In this work, we focus on the class of *biclique-free* graphs, which has received considerable attention from algorithmic perspectives (Aboulker et al., 2023; Eiben et al., 2019; Fabianski et al., 2019; Koana et al., 2022; Lokshtanov et al., 2018; Telle and Villanger, 2019). For $i, j \in \mathbb{N}$, let $K_{i,j}$ denote the complete bipartite graph on i vertices on one side and j vertices on the other side. We consider the following problem.

Biclique Free Vertex Deletion (BFVD)

Input: An undirected graph G and $i, j, k \in \mathbb{N}$, $i \leq j$.
Question: Does there exist a subset $V' \subseteq V$ with $|V'| \leq k$ such that $G - V'$ does not contain any $K_{i,j}$ as a (not necessarily induced) subgraph?

*Work was done while affiliated with TU Berlin. Supported by the DFG Project DiPa, Ni 369/21.

ISSN 1365–8050

Note that we consider i and j to be part of the input, that is, they are not to be treated as a constant. Hence, BFVD is a generalization of the BOUNDED DEGREE DELETION problem, defined as follows.

BOUNDED DEGREE DELETION (BDD)

Input: An undirected graph G and $r, k \in \mathbb{N}$

Question: Is there a subset $V' \subseteq V$ with $|V'| \leq k$ such that each vertex in $G - V'$ has degree at most r?

Note that an instance (G, k, r) of BDD is a yes-instance, if and only if the instance $(G, 1, r+1, k)$ of BFVD is a yes-instance. Furthermore, note that the special case $r = 0$ of BDD, that is, BFVD with $i = j = 1$, is VERTEX COVER.

BDD appears in the field of computational biology (Fellows et al., 2011). Its dual, the so-called k-PLEX DELETION problem, is a clique relaxation problem that finds many applications in social network analysis (Balasundaram et al., 2011; McClosky and Hicks, 2012; Moser et al., 2012; Seidman and Foster, 1978). Hence, it is not surprising that its computational complexity has been studied extensively in the last two decades (Balasundaram et al., 2010; Betzler et al., 2012; Bodlaender and van Antwerpen-de Fluiter, 2001; Chen et al., 2010; Dessmark et al., 1993; Komusiewicz et al., 2009; Nishimura et al., 2005; Seidman and Foster, 1978). Its parameterized complexity has been studied as well: BDD is fixed-parameter tractable (FPT) with respect to $r + k$ (Fellows et al., 2011; Moser et al., 2012; Nishimura et al., 2005), but W[2]-hard with respect to k (Fellows et al., 2011). As for structural parameterizations, BDD is known to be FPT with respect to the degeneracy plus k (Raman et al., 2008) and with respect to the feedback edge number and to the treewidth plus r (Betzler et al., 2012). Recently, Ganian et al. (2021) showed that BDD is W[1]-hard when parameterized by the feedback vertex number or the treedepth,[i] but FPT when parameterized by the treecut width (Marx and Wollan, 2014). Lampis and Vasilakis (2023) proved some fine-grained conditional lower bounds for BDD with respect to treewidth and vertex cover number.

Allowing i and j to be part of the input makes BFVD challenging to solve. Deciding whether the input graph G is free of bicliques $K_{i,j}$ is challenging on its own. In fact, the problem of determining whether G contains a biclique is NP-hard and W[1]-hard with respect to $i + j$. Thus, BFVD is coNP-hard and coW[2]-hard for $i + j$ even if $k = 0$ (Lin, 2018). On degenerate graphs however, one can efficiently enumerate all maximal bicliques (Eppstein, 1994). For this reason, and in order to see which results for BDD also hold for its generalization, we study the computational tractability of BFVD with respect to structural graph parameters.

Our results. We first show in Section 2 that BFVD can be solved in $\mathcal{O}^*(2^{\mathcal{O}(\mathrm{vc} \cdot k)})$ time, where vc is the minimum vertex cover size of G. This paves the way for the algorithms presented in Sections 3 and 4. Using the $\mathcal{O}^*(2^d)$-time algorithm of Eppstein (1994), where d is the degeneracy of G, to enumerate all maximal bicliques, we show that each vertex and edge not part of any biclique $K_{i,j}$ can be identified (and deleted) in time $\mathcal{O}^*(4^d)$. When every edge is part of some biclique $K_{i,j}$, the set of vertices that appear in the smaller side of some biclique $K_{i,j}$ form a vertex cover. In Section 3, we show that BFVD can be solved in $\mathcal{O}^*(2^{\mathcal{O}(dk^2)})$ time. The algorithm takes a win-win approach: If there are not many vertices that appear in the smaller side of a biclique, then we use the aforementioned $\mathcal{O}^*(2^{\mathcal{O}(\mathrm{vc} \cdot k)})$-time algorithm. Otherwise, we can find a set of vertices which has a nonempty intersection with every solution. Following the same approach albeit with a more refined analysis, we develop in Section 3 an algorithm for

[i] see Section 2 for a definition of the parameters

BFVD running in $\mathcal{O}^*(2^{\mathcal{O}(k^2+\mathrm{fvn}\cdot k)})$ time. That actually implies that BFVD is fixed-parameter tractable for fvn when $i \geq 2$ since an instance with $k \geq \mathrm{fvn}$ is a yes-instance. In contrast, we show in Section 5 that BFVD is W[1]-hard for every $i \in \mathbb{N}$ when parameterized by treedepth. To the best of our knowledge, BFVD is the first problem shown to be FPT for the feedback vertex number but W[1]-hard for the treedepth. Incidentally, there are several problems that behave in the opposite way, i. e., are FPT for the treedepth but W[1]-hard for the feedback vertex number such as MIXED CHINESE POSTMAN (Gutin et al., 2016), GEODETIC SET (Kellerhals and Koana, 2022), and LENGTH-BOUNDED CUT (Bentert et al., 2022; Dvořák and Knop, 2018). Finally, we show in Section 6 that BFVD admits a polynomial kernel for the feedback edge number, strengthening the fixed-parameter tractability result of Betzler et al. (2012).

2 Preliminaries

Let $\mathbb{N}$ be the set of positive integers. For $n \in \mathbb{N}$, let $[n] := \{1, 2, \ldots, n\}$.

Graphs. For standard graph terminology, we refer to Diestel (2017). For a graph G, let $V(G)$ denote its vertex and let $E(G)$ denote its edge set. Let $X \subseteq V(G)$ be a vertex set. We denote by $G[X]$ the subgraph induced by X and let $G - X$ be $G[V(G) \setminus X]$. For an edge set $F \subseteq E(G)$, we denote by $G \setminus F$ the graph $(V(G), E \setminus F)$. Let $N_G(X) = \{y \mid x \in X, xy \in E(G)\} \setminus X$ and $N_G[X] = N_G(X) \cup X$. We drop the subscript $\cdot_G$ when not ambiguous. For simpler notation, we sometimes use x for $\{x\}$.

For fixed $i \leq j \in \mathbb{N}$, we say that a pair (S, T) of disjoint vertex sets is a biclique if $|S| = i$, $|T| = j$, and $st \in E(G)$ for every $s \in S$ and $t \in T$. We refer to S as the *smaller side* and T as the *larger side*. If $|\bigcap_{s \in S} N(s)| \geq j$, then let $(S, \cdot)$ denote an arbitrary biclique (S, T) with $|T| = j$. Let $\mathcal{S}_G$ be the collection of smaller sides of all bicliques $K_{i,j}$ of G and let $\mathsf{ss}(G) = |\bigcup_{S \in \mathcal{S}_G} S|$. Whenever i and j are clear from context, we allow ourselves to just call a $K_{i,j}$ just *biclique*.

Graph parameters. The *vertex cover number* $\mathrm{vc}(G)$ is the size of a smallest set $V' \subseteq V(G)$ such that $G - V'$ is edgeless. A set $F \subseteq E(G)$ is a *feedback edge set* if $G \setminus F$ is a forest. The *feedback edge number* $\mathrm{fen}(G)$ is the size of a smallest such set. A set $D \subseteq V(G)$ is a *feedback vertex set* if $G - D$ is a forest. The *feedback vertex number* $\mathrm{fvn}(G)$ is the size of a smallest such set.

For a graph G, a *tree decomposition* is a pair (T, B), where T is a tree and $B\colon V(T) \to 2^{V(G)}$ such that (i) for each edge $uv \in E(G)$ there exists $x \in V(T)$ with $u, v \in B(x)$, and (ii) for each $u \in V(G)$ the set of nodes $x \in V(T)$ with $v \in B(x)$ forms a nonempty, connected subtree in T. The *width* of (T, B) is $\max_{x \in V(T)}(|B(x) - 1|)$. The *treewidth* $\mathrm{tw}(G)$ of G is the minimum width of all tree decompositions of G.

The *treedepth* of a connected graph G is defined as follows (Nešetřil and Ossona de Mendez, 2006). Let T be a rooted tree with vertex set $V(G)$, such that if $uv \in E(G)$, then u is either an ancestor or a descendant of v in T, i.e., the path from u to v in T does not contain the root as an inner vertex. We say that G is *embedded* in T. The *depth* of T is the number of vertices in a longest path in T from the root to a leaf. The *treedepth* $\mathrm{td}(G)$ of G is the minimum t such that there is a rooted tree of depth t in which G is embedded.

See Figure 1 for the relationship between parameters. Throughout this paper, for any of the parameters $\mathrm{x}(G)$ introduced above, we allow ourselves to simply write x if the graph G is clear from context.

Parameterized complexity. A *parameterized problem* is a subset $L \subseteq \Sigma^* \times \mathbb{N}$ over a finite alphabet Σ. Let $f\colon \mathbb{N} \to \mathbb{N}$ be a computable function. A parameterized problem L is *fixed-parameter tractable (in*

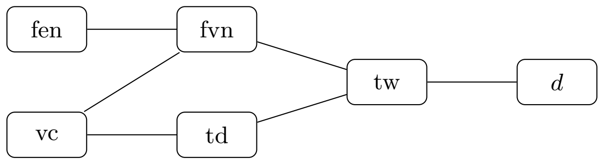

Fig. 1: A Hasse diagram of parameters we study in this paper. An edge from x (left) to y (right) indicates that $x(G) \geq y(G) - 1$ for every graph G.

FPT*)* with respect to k if $(I, k) \in L$ is decidable in $f(k) \cdot |I|^{\mathcal{O}(1)}$ time. A *kernel* for this problem is an algorithm that takes the instance (I, k) and outputs a second instance (I', k') such that (i) $(I, k) \in L$ if and only if $(I', k') \in L$ and (ii) $|I'| + k' \leq f(k)$ for a computable function f. The *size* of the kernel is f. We call a kernel *polynomial* if f is a polynomial. To show that a problem L is (presumably) not in FPT, one may use a *parameterized reduction* from a problem that is hard for the class W[1] to L. A parameterized reduction from a parameterized problem L to another parameterized problem L' is a function that acts as follows: For functions f and g, given an instance (I, k) of L, it computes in $f(k) \cdot |I|^{\mathcal{O}(1)}$ time an instance (I', k') of L' so that $(I, k) \in L$ if and only if $(I', k') \in L'$ and $k' \leq g(k)$. For more details on parameterized complexity, we refer to the standard monographs (Cygan et al., 2015; Downey and Fellows, 2013; Niedermeier, 2006).

Preliminary results. Our algorithms will use the following simple reduction rule.

Rule 1. Delete a vertex or edge that is not part of any biclique $K_{i,j}$.

Note, however, that to apply Rule 1 exhaustively, one has to enumerate all bicliques efficiently. Although it may take $\Omega(2^n)$ time to enumerate all bicliques (S, T), it is known that all *maximal* bicliques can be listed in $\mathcal{O}^*(2^d)$ time in d-degenerate graphs (Eppstein, 1994). In fact, we can enumerate all vertex sets that comprise the smaller sides of all bicliques $K_{i,j}$, which we denote by $\mathcal{S}_G$, in $\mathcal{O}^*(4^d)$ time as follows: First we enumerate all maximal bicliques. At least one side S' of each maximal biclique is of size at most $d + 1$, since a d-degenerate graph does not contain a biclique $K_{d+1,d+1}$ (note that every vertex has degree $d + 1$ in $K_{d+1,d+1}$). Hence, we may assume that $i \leq d$. For each subset $S \subseteq S'$ of size i (there are $\binom{|S'|}{i} \leq 2^{d+1}$ many), we add S to $\mathcal{S}_G$ if S is not in $\mathcal{S}_G$ and $|\bigcap_{s \in S} N(s)| \geq j$. With $\mathcal{S}_G$ at hand, we can apply Rule 1 exhaustively with polynomial-time overhead.

After applying Rule 1 exhaustively, at least one endpoint of each edge appears in the smaller side of some biclique, resulting in the following lemma.

Lemma 1. *If Rule 1 has been applied exhaustively, then* $\bigcup_{S \in \mathcal{S}_G} S$ *is a vertex cover of* G.

Next, we show that BFVD is FPT when parameterized by the vertex cover number $\mathrm{vc}(G)$.

Proposition 2. BFVD *can be solved in* $\mathcal{O}^*(2^{\mathcal{O}(\mathrm{vc} \cdot k)})$ *time.*

Proof: For an instance $\mathcal{I} = (G, k, i, j)$ of BFVD, let X be a vertex cover of G. Note that $G - X$ is biclique-free, thus if $\mathrm{vc} \leq k$, we return X as a solution and are done. So suppose that $\mathrm{vc} > k$ and let V' be a hypothetical solution of $\mathcal{I}$. First, our algorithm guesses the subset $X' = V' \cap X$. For the remaining vertices $v \in V' \setminus X$, we know that $N(v) \subseteq X$, and we guess the neighborhood of each of the at most k vertices in $V' \setminus X$. Let $\mathcal{N}$ be the (multi-)set of the guessed neighborhoods. Note that there are at most $2^{|X|}$ choices for the first guess and at most $2^{|X|}$ choices for each of the neighborhoods, resulting in at most

$2^{\mathcal{O}(\mathrm{vc}\cdot k)}$ choices. We arbitrarily choose a distinct vertex $v \in V(G) \setminus X$ with $N(v) = Y$ for each $Y \in \mathcal{N}$ and we delete it from G. We also delete X' from the graph. If the resulting graph has no biclique $K_{i,j}$, which can be determined in $\mathcal{O}^*(2^d) = \mathcal{O}^*(2^{\mathrm{vc}})$ time, then we conclude that $\mathcal{I}$ is a `yes`-instance. □

Lemma 1 and Proposition 2 imply that BFVD is fixed-parameter tractable with respect to $\mathsf{ss}(G)$ on d-degenerate graphs. This fact will play an important role in the algorithm presented in Sections 3 and 4.

Finally, we show that BFVD is FPT when k, i, j, d are part of the parameter. Our algorithm essentially solves an instance of HITTING SET in which every set has size at most $i + j$.

Proposition 3. BFVD *can be solved in* $\mathcal{O}^*(4^d \cdot (i + j)^k)$ *time.*

Proof: We solve an instance $\mathcal{I} = (G, k, i, j)$ of BFVD recursively as follows: If there is a biclique (S, T) (which can be found in $\mathcal{O}^*(4^d)$ time), then $\mathcal{I}$ is a `yes`-instance if and only if $(G - v, k - 1, i, j)$ is a `yes`-instance for some $v \in S \cup T$. The search tree has depth at most k and each node has at most $i + j$ children, and thus the running time is $\mathcal{O}^*(4^d \cdot (i + j)^k)$. □

We remark that BFVD is unlikely to be FPT for $i + j + k$, since it is coW[1]-hard for $i + j$ when $k = 0$, as mentioned in the introduction.

3 FPT with degeneracy and solution size

In this section, we show that BFVD can be solved in $\mathcal{O}^*(2^{\mathcal{O}(dk^2)})$ time on graphs with degeneracy d, extending the known fixed-parameter tractability of BDD (Raman et al., 2008). Essentially, our algorithm considers two cases based on the value of $\mathsf{ss}(G)$. If $\mathsf{ss}(G)$ is sufficiently small, then we invoke the algorithm of Proposition 2. Otherwise, we aim to find a few vertices that intersect a solution. To find such vertices, we use the following lemma of Alon and Gutner (2009), which has been also applied to show fixed-parameter tractability of several domination problems (including BDD) (Alon and Gutner, 2009; Golovach and Villanger, 2008; Raman et al., 2008).

Lemma 4 ((Alon and Gutner, 2009))**.** *Let X be a set of at least $(4d + 2)k$ vertices. Then there are at most $(4d + 2)k$ vertices that are adjacent to at least $|X|/k$ vertices of X.*

Using Lemma 4, we will show that a set of $\mathcal{O}(dk)$ vertices that intersects a hypothetical solution can be found in polynomial time whenever $\mathcal{S}_G$ is sufficiently large (Lemma 6). Recall that $\mathcal{S}_G$ is the collection of smaller sides of all bicliques $K_{i,j}$. The proof of Lemma 6 relies on the following lemma.

Lemma 5. *Let $\mathcal{I} = (G, k, i, j)$ be a `yes`-instance of* BFVD. *Let $\mathcal{X} \subseteq \mathcal{S}_G$ be a nonempty collection of smaller sides of bicliques and let $X = \bigcup_{X' \in \mathcal{X}} X'$. Suppose that V' is a solution of $\mathcal{I}$ with $V' \cap X = \emptyset$. Then there exists a vertex $v \in V'$ which has at least $|X|/k$ neighbors in X.*

Proof: Assume to the contrary that every vertex v in V' has less than $|X|/k$ neighbors in X, i.e., $|X \cap N(v)| < |X|/k$. Then we have $X \setminus N(V') \neq \emptyset$ since

$$|X \setminus N(V')| \geq |X| - \Big| \bigcup_{v \in V'} (X \cap N(v)) \Big| \geq |X| - \sum_{v \in V'} |X \cap N(v)| > 0.$$

Choose any $u \in X \setminus N(V')$. By the definition of X, there exists a biclique (S, T) such that $u \in S$ and $S \subseteq X$. Since $V' \cap X = \emptyset$, the solution V' does not intersect S. It does not intersect T either, since every vertex in T is adjacent to u. Thus, there remains a biclique $K_{i,j}$ in $G - V'$, a contradiction. □

Lemma 6. *Let $\mathcal{I} = (G, k, i, j)$ be an* `yes`*-instance of* BFVD*. If* $\mathsf{ss}(G) > (4d+2)k$*, then we can find in polynomial time a set W of at most $(8d+4)k+i$ vertices such that $V' \cap W \neq \emptyset$ for every solution V' of $\mathcal{I}$.*

Proof: We first find an inclusion-wise minimal subcollection $\mathcal{X} \subseteq \mathcal{S}_G$ of smaller sides of bicliques such that $|\bigcup_{X' \in \mathcal{X}} X'| \geq (4d+2)k$. This can be done in polynomial time using a simple greedy algorithm. Let $X = \bigcup_{X' \in \mathcal{X}} X'$. Note that $|X| \leq (4d+2)k + i$ since otherwise the deletion of arbitrary $X' \in \mathcal{X}$ from $\mathcal{X}$ gives us another desired subcollection, contradicting the minimality of $\mathcal{X}$. We will include X into W. Thus, $V' \cap W \neq \emptyset$ holds if $V' \cap X \neq \emptyset$. If $V' \cap X = \emptyset$, then by Lemma 5, there exists a vertex $v \in V'$ which has at least $|X|/k$ neighbors in X. Let U be the set of vertices u with at least $|X|/k$ neighbors in X. By Lemma 4, $|U| \leq (4d+2)k$. Thus, the lemma holds for $W = X \cup U$. □

Finally, we show that BFVD is FPT with respect to $k + d$ using Lemma 6.

Theorem 7. BFVD *can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(dk^2)})$.*

Proof: Given an instance (G, k, i, j) of BFVD, we first apply Rule 1 exhaustively. If $i > d$, then we obtain a trivial `yes`-instance since G does not have $K_{d+1,d+1}$. We consider two cases. Suppose first that $\mathsf{ss}(G) \leq (4d+2)k$. Since $\bigcup_{S \in \mathcal{S}_G} S$ is a vertex cover of G by Lemma 1, we can solve the instance using the algorithm of Proposition 2 in $\mathcal{O}^*(2^{\mathcal{O}(dk^2)})$ time. Otherwise, we have $\mathsf{ss}(G) > (4d+2)k$. In this case, we can find in polynomial time a set W of size at most $(8d+4)k+i \in \mathcal{O}(dk)$ vertices that intersects every solution by Lemma 6. For every vertex $w \in W$, we recursively solve the instance $(G - w, k-1, i, j)$ until we have a trivial instance or $\mathsf{ss}(G) \leq (4d+2)k$. The running time is bounded by $\mathcal{O}^*(2^{\mathcal{O}(dk^2)})$. □

4 FPT with feedback vertex number

In the previous section, we have seen an $\mathcal{O}^*(2^{\mathcal{O}(dk^2)})$-time algorithm for BFVD. We present an algorithm for BFVD running in $\mathcal{O}^*(2^{\mathcal{O}(k^2 + \mathrm{fvn} \cdot k)})$ time in this section:

Theorem 8. *For $i \geq 2$,* BFVD *can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k^2 + \mathrm{fvn} \cdot k)})$ time.*

For $i \geq 2$, any instance (G, k, i, j) with $k \geq \mathrm{fvn}$ is a `yes`-instance, since a forest does not contain any biclique $K_{i,j}$. Thus, we have the following corollary:

Corollary 9. *For $i \geq 2$,* BFVD *can be solved in $\mathcal{O}^*(2^{\mathcal{O}(\mathrm{fvn}^2)})$ time.*

We remark that, as the degeneracy of a graph is at most $\mathrm{fvn} + 1$, this is faster than the running time of the algorithm derived analogously from Theorem 7, which is $\mathcal{O}^*(2^{\mathcal{O}(\mathrm{fvn}^3)})$.

Algorithm 1 provides an overview of the algorithm that shows Theorem 8. In a nutshell, we identify several cases that are efficiently solvable. If none of the cases apply, then $\mathsf{ss}(G) \in \mathcal{O}(k + \mathrm{fvn})$ holds, and we can use the algorithm of Proposition 2. Let V' denote a hypothetical solution and let D be a minimum feedback vertex set. We first guess the intersection $D' = V' \cap D$, which we delete from the graph. Let $R \subseteq V(G) \setminus D$ be the set of vertices whose closed neighborhood contains at least three vertices in $\bigcup_{S \in \mathcal{S}_G} S$. As we show later in Lemma 11, we can immediately conclude that we have a `no`-instance if $|R| > 3k$ (Line 5). Again, we guess the intersection $R' = V' \cap R$ to be deleted from the graph. If more than $2k$ vertices in the forest $F = G - D$ remain in $\bigcup_{S \in \mathcal{S}_G} S$, then we can conclude that the instance has no solution (Line 7), as we show in Lemma 12.

Algorithm 1: The algorithm for Theorem 8. We assume that Rule 1 is exhaustively applied throughout.

Input : A graph G, integers $i \leq j$, k, and a feedback vertex set D.
1 **if** $j \leq \text{fvn} + 1$ **then return** the result of the algorithm of Proposition 3.
2 **guess** $D' \subseteq D$. Remove D' from G and D, set $k \leftarrow k - |D'|$.
3 $F \leftarrow G - D$. Root F arbitrarily.
4 $R \leftarrow \{v \in V(F) \mid |N_F[v] \cap \bigcup_{S \in \mathcal{S}_G} S| \geq 3\}$.
5 **if** $|R| > 3k$ **then return** `no`. (Lemma 11)
6 **guess** $R' \subseteq R$, $|R'| \leq k$. Remove R' from G and F, set $k \leftarrow k - |R'|$.
7 **if** $|V(F) \cap \bigcup_{S \in \mathcal{S}_G} S| > 2k$ **then return** `no`. (Lemma 12)
8 **return** the result of the algorithm of Proposition 2.

The following observation that at most one vertex of F appears in the smaller side of a biclique becomes crucial to establish the correctness of Algorithm 1.

Observation 10. *If $j > \text{fvn} + 1$, then the smaller side S of every biclique (S, T) contains at most one vertex of $V(F)$.*

Proof: If $j > \text{fvn} + 1$, then for each biclique (S, T), there are two vertices $u, v \in T \cap V(F)$. If there are two vertices in $S \cap V(F)$, then they induce a cycle with u and v in the forest, a contradiction. ☐

The following lemma shows that Line 5 is correct. Since we delete the intersection $D' = V' \cap D$ from the graph in Line 2, we may assume that $V' \subseteq V(F)$.

Lemma 11. *If $|R| > 3k$ in Line 5, then every set $V' \subseteq V(F)$ that intersects every $K_{i,j}$ contains more than k elements.*

Proof: Partition R into three sets R_0, R_1, R_2 such that v is in R_δ if the distance from v to the root of the same component is δ modulo 3. At least one of the partitions, say R_0, contains more than k elements. By the definition of R, we have $|N_F[v] \cap \bigcup_{S \in \mathcal{S}_G} S| \geq 3$ for every $v \in R$. Note that $N_F[v]$ consists of v itself, its parent (if it exists), and its child(ren). It follows that v has a child $q_v \in \bigcup_{S \in \mathcal{S}_G} S$. Let (S_v, T_v) be an arbitrary biclique $K_{i,j}$ with $q_v \in S_v$ and let $U_v = S_v \cup T_v$. By Observation 10, we have $S_v \cap V(F) = \{q_v\}$. Thus, $T_v \cap V(F) \subseteq N_F(q_v)$. As $q_v \in R_1$, we have $N_F(q_v) \cap R_0 = \{v\}$, and all remaining neighbors of q_v are in R_2. Now pick $v' \in R_0$ with $v' \neq v$ and define $S_{v'}$, $T_{v'}$, $U_{v'}$ and $q_{v'}$ analogously. Then $N_F[q_v] \cap N_F[q_{v'}] = \emptyset$ as $v \neq v'$ and $q_v \neq q_{v'}$; the remaining vertices in the neighborhoods are children of either q_v or $q_{v'}$ and thus cannot be equal either. Consequently $U_v \cap V(F)$ and $U_{v'} \cap V(F)$ are disjoint. Hence, a set $V' \subseteq V(F)$ intersecting every biclique contains at least one vertex of $U_v \cap V(F)$ for every vertex in $v \in R_0$. Thus, $|V'| \geq |R_0| > k$. ☐

Next, we show that Line 7 is correct. In Line 6, we delete the vertices of R included in the hypothetical solution V'. We thus may assume that V' is disjoint from R.

Lemma 12. *If $|V(F) \cap \bigcup_{S \in \mathcal{S}_G} S| > 2k$ in Line 7, then every set $V' \subseteq V(F) \setminus R$ that intersects every $K_{i,j}$ contains more than k elements.*

Proof: Suppose that there exists a set $V' \subseteq V(F) \setminus R$ that intersects every $K_{i,j}$ of size at most k. By the definition of R, every vertex $v' \in V'$ has $|N_F[v'] \cap \bigcup_{S \in \mathcal{S}_G} S| \leq 2$. As $|V(F) \cap \bigcup_{S \in \mathcal{S}_G} S| > 2k \geq 2|V'|$, we have

$$\Big|\Big(V(F) \cap \bigcup_{S \in \mathcal{S}_G} S\Big) \setminus N[V']\Big| \geq \Big|V(F) \cap \bigcup_{S \in \mathcal{S}_G} S\Big| - \sum_{v' \in V'} \big|N_F[v'] \cap \bigcup_{S \in \mathcal{S}_G} S\big| > 0.$$

This implies the existence of a vertex $v \in (V(F) \cap \bigcup_{S \in \mathcal{S}_G} S) \setminus N[V']$. Let (S_v, T_v) be an arbitrary biclique with $v \in S_v$ and let $U_v = S_v \cup T_v$. By Observation 10, we have $U_v \cap V(F) \subseteq N_F[v]$. Since $v \notin N[V']$, $V' \subseteq V(F)$ does not intersect U_v, a contradiction. □

Finally, we analyze the running time of Algorithm 1.

Lemma 13. *Algorithm 1 runs in time $\mathcal{O}^*(2^{\mathcal{O}(k^2 + \mathrm{fvn} \cdot k)})$.*

Proof: In Algorithm 1, we guess D' and R'. There are $2^{\mathcal{O}(\mathrm{fvn})}$ choices for D' and 2^{3k} choices for R', amounting to $2^{\mathcal{O}(k+\mathrm{fvn})}$ choices. Since we have $\mathsf{ss}(G) \leq |V(F) \cap \bigcup_{S \in \mathcal{S}_G} S| + |D \cap \bigcup_{S \in \mathcal{S}_G} S| \leq 3k + \mathrm{fvn}$ in Line 8 and $\bigcup_{S \in \mathcal{S}_G} S$ is a vertex cover of G by Lemma 1, the algorithm of Proposition 2 runs in time $\mathcal{O}^*(2^{\mathcal{O}(k^2 + \mathrm{fvn} \cdot k)})$. We spend $\mathcal{O}^*(4^{\mathrm{fvn}})$ time elsewhere; hence Algorithm 1 runs in the claimed time. □

The correctness of Algorithm 1 follows from Propositions 2 and 3 and Lemmas 11 and 12 and it runs in time $\mathcal{O}^*(2^{\mathcal{O}(k^2 + \mathrm{fvn} \cdot k)})$ by Lemma 13. This proves Theorem 8.

5 Parameterized hardness

Ganian et al. (2021) show that BFVD is W[1]-hard with respect to the treedepth of the input graph if $i = 1$. We show that this holds true for every fixed value of i.

Theorem 14. *For every fixed i,* BFVD *is* W[1]*-hard when parameterized by the treedepth.*

Proof: We reduce from BDD, which is W[1]-hard when parameterized by treedepth (Ganian et al., 2021). Given an instance (G, k, r) of BDD, we construct an instance (G', k, i, j) as follows. We set $j = n + r + 1$, where n is the number of vertices in G. We will assume that $n > i$. For the construction of G', we start with a copy of G. For every $v \in V(G)$, we introduce $i - 1$ vertices $S_v = \{s_v^1, \ldots, s_v^{i-1}\}$ and n vertices $T_v = \{t_v^1, \ldots, t_v^n\}$ and add edges such that $S'_v = \{v\} \cup S_v$ and T_v form a biclique $K_{i,n}$. Moreover, for every edge $uv \in E(G)$, we add an edge between u and s_v for every $s_v \in S_v$.

Suppose that (G, k, r) has a solution V'. We claim that V' is also a solution of (G', k, i, j). Suppose to the contrary that $G - V'$ has a biclique $K_{i,j}$. Then, its smaller side is S'_v for some vertex $v \in V(G)$ and its larger side is a subset of $(N_G(v) \setminus V') \cup T_v$. To see why, observe that $V(G) \cup \bigcup_{v \in V(G)} S_v$ constitutes the set of vertices of degree at least $n > i$ and that two vertices have at least n common neighbors in G' if and only if they belong to the same S'_v. In particular, it holds that $v \notin V'$. Since every vertex in $V \setminus V'$ has degree at most r in $G - V'$, we have $|N_G(v) \setminus V'| \leq r$ and thus $|(N_G(v) \setminus V') \cup T_v| \leq n + r$, a contradiction.

Conversely, suppose that V' is a solution of (G', k, i, j). We claim that the set $V'' = \{v \in V(G) \mid (S'_v \cup T_v) \cap V' \neq \emptyset\}$ is a solution of (G, k, r). Note that $|V''| \leq |V'| \leq k$. Suppose that there exists a vertex $v \in V(G) \setminus V'$ of degree greater than r in $G - V''$. Then, S'_v is of size i and it has at least $|(N_G(v) \setminus V') \cup T_v| \geq j$ common neighbors. We thus conclude that $G - V''$ is a solution.

Finally, we show that $\mathrm{td}(G') \leq i \cdot \mathrm{td}(G) + 1$ by providing a rooted tree T' of depth $\mathrm{td}(G')$ in which G' is embedded. The tree is based on a rooted tree T of depth $\mathrm{td}(G)$ in which G is embedded. We replace every vertex $v \in V(T)$ with a path consisting of the vertices in S'_v and attach the children of v in T to the lowermost (furthest from the root) vertex in S'_v. Then we add each $t_v \in T_v$ as a leaf to the lowermost vertex in S'_v. Note that any ancestor $u \in V(G)$ of v is now ancestor of all vertices in S'_v, and each vertex in S'_v is ancestor of each vertex in T_v; thus G' is embedded in T'. As we replace every vertex with a path of length i, and attach at most one child at the bottom of the path, the depth of T' is at most $i \cdot \mathrm{td}(G) + 1$. □

6 Polynomial kernel with respect to feedback edge number

In this section, we show that BFVD admits a polynomial kernel when parameterized by the feedback edge number fen.

Theorem 15. BFVD *admits a kernel of size $\mathcal{O}(\mathrm{fen}^2)$ for $i = 1$ and $\mathcal{O}(\mathrm{fen})$ for $i \geq 2$.*

We start with the case $i = 1$. Then BFVD is equivalent to BOUNDED-DEGREE DELETION (BDD) as there is a trivial parameter-preserving reduction (set $r = j - 1$, where r is the degree bound of BDD and j is the size of one of the biclique sides). It is known that BDD is fixed-parameter tractable for fen (Betzler et al., 2012). We strengthen their result proving the existence of a polynomial kernel.

To develop a kernelization algorithm, we will work with the following generalization of BDD.

WEIGHTED BOUNDED-DEGREE DELETION (WBDD)

Input: An undirected graph G, two integers $k, r \in \mathbb{N}$, and weights $w \in \mathbb{N}^{V(G)}$.
Question: Does there exist a subset $V' \subseteq V(G)$ with $|V'| \leq k$ such that each vertex $v \in V(G) \setminus V'$ has degree at most $r - w_v$ in $G - V'$?

Herein, by w_v we denote the weight of a vertex v. Note that BDD is a special case of WBDD, where $w_v = 0$ for each $v \in V$.

We use the weights in the following manner: Suppose that for an instance of WBDD, we identify a vertex v which can be "avoided", that is, there is a solution V' with $v \notin V'$. Then we can simplify the instance as follows: delete v and increase the weight of every neighbor of v by one.

To show that BDD admits a kernel of size $\mathcal{O}(\mathrm{fen}^2)$, we first show that WBDD has a kernel of size $\mathcal{O}(\mathrm{fen})$ for constant r. We then show how, given a WBDD instance of size $\mathcal{O}(\mathrm{fen})$, we can transform it into a BDD instance of size $\mathcal{O}(\mathrm{fen}^2)$.

Linear kernel for WBDD. As a first step to obtain a linear kernel for WBDD, we apply reduction rules based on $\deg(v)$ and w_v. We first observe that our problem treats a vertex v the same whenever $\deg(v) + w_v \leq r$. Hence, we may set the weight w_v of such vertices to $r - \deg(v)$.

Rule 2. If $\deg(v) + w_v < r$, then increase w_v by one.

Next, if a weight of a vertex is too high, then it must be in any solution.

Rule 3. If $w_v > r$, then delete v and decrease k by one.

After applying these two reduction rules, we have $r - \deg(v) \leq w_v \leq r$ for every vertex v. In particular, we have $w_v = r$ for every isolated vertex v, which can be deleted.

Rule 4. Let $v \in V$ be an isolated vertex of G with $w_v = r$. Then, delete v.

For a degree-one vertex v, we have $w_v = r - 1$ or $w_v = r$. In either case, it does not make sense to take v into the solution, as deleting its neighbor affects the degrees of at least as many vertices.

Rule 5. Let $v \in V$ be a vertex of G with $N(v) = \{u\}$. If $w_v = r - 1$, then delete v and increase w_u by one. If $w_v = r$, then delete both v and u, and decrease k by one.

Lemma 16. *Rules 2 to 5 are correct and can be applied exhaustively in $\mathcal{O}(n + m)$ time.*

Proof: Suppose $V' \subseteq V(G)$ is a solution for an instance of WBDD and consider a vertex $v \in V(G)$. Suppose that $\deg(v) + w_v \leq r$. Then, V' remains a solution if we replace w_v with a weight w'_v, with $0 \leq w'_v \leq r - \deg(v)$. Hence, Rule 2 is correct. Suppose next that $w_v > r$. Then any solution V' must contain v. Hence, $G - \{v\}$ contains a solution of size $k - 1$ if and only if G contains a solution of size k, and Rule 3 is correct. Suppose now that $N(v) = \{u\}$. If $w_v = r$, then either $u \in V'$ or $v \in V'$. As the choice of $u \in V'$ decreases the degree of at least as many vertices as $v \in V'$, we may always pick u into V'. Suppose that $w_v = r - 1$. If $v \in V'$, then $(V' \setminus \{v\}) \cup \{u\}$ is also a valid solution by the same argument; Otherwise, V' remains a solution. Conversely, any solution for the resulting instance is a solution for the original instance because we increase the weight w_u by one; thus Rule 5 is correct. The correctness of Rule 4 is obvious.

To apply the reduction rules exhaustively, we have to test for each vertex if one of the reduction rules can be applied and repeat this test for all vertices where at least one neighbor got deleted by applying one of the rules. This can be done in $\mathcal{O}(n + m)$ time by checking the weight and the degree of each vertex on a list of vertices that still need to be tested. If a vertex gets deleted, then all neighbors that are not on the list already need to be added again. The actual exhaustive application of the rules requires $\mathcal{O}(n + m)$ time, since deleting all vertices while still maintaining a correct graph representation is possible in this time. Combining the steps leads to $\mathcal{O}(n + m)$ time in total. □

We will henceforth assume that Rules 2 to 5 have been exhaustively applied.

To obtain a linear kernel for WBDD, we use the following folklore result. We call a path *maximal* if both of its endpoints have degree at least three and all inner vertices have degree exactly two. We call a cycle *maximal* if at most one of its vertices has degree at least three.

Lemma 17 (See e.g., (Epstein et al., 2015; Kellerhals and Koana, 2022))**.** *Let G be a graph in which each vertex has degree at least two. Then, the number of vertices of degree at least three is at most* $2\mathrm{fen} - 2$. *Moreover, the number of maximal paths and cycles in G is at most* $3\mathrm{fen} - 3$.

By Lemma 17, the number of vertices of degree at least three is at most $2\mathrm{fen} - 2$. It remains to bound the length of maximal paths and cycles in which each vertex has degree two. We introduce further notation. For a (vertex-) weighted graph (G, w), let $\mathrm{opt}(G, w)$ denote the minimum integer k such that (G, k, r, w) is a `yes`-instance of WBDD. If G is a path, then $\mathrm{opt}(G, w)$ is linear-time computable by a trivial adaptation of Rules 2 to 5.

Our algorithm works as follows: If the graph contains a sufficiently long path P of degree-two vertices, then we replace it with another weighted path P' of shorter length. The replacement path P' should behave analogously to P in the context of WBDD. Our key finding is that the *characteristic matrix* of weighted degree-two paths determines the behavior of weighted paths. Intuitively, the characteristic matrix captures the increase in the optimal solution size when a subset of the four outermost vertices (i.e., the endpoints and each of their neighbors) is included.

Definition 18. For an integer $\ell \geq 5$, let $\mathcal{P}_\ell$ denote the collection of weighted paths (P, w) on ℓ vertices $v_1, \ldots, v_\ell$ such that $w_{v_1} = w_{v_\ell} = r - 1$ and $w_{v_i} \in \{r - 2, r - 1, r\}$ for each $i \in [2, \ell - 1]$. For a weighted path $(P, w) \in \mathcal{P}_\ell$, the *characteristic matrix* of (P, w) is a 3×3 matrix $M(P, w)$ such that $M(P, w)_{x,y} = s_{x,y} - \mathrm{opt}(P, w)$, where $s_{x,y}$ is the minimum size of a vertex set S such that

(i) $v_1 \in S$ if $x = 1$, $v_2 \in S$ and $v_1 \notin S$ if $x = 2$ and $v_1, v_2 \notin S$ if $x = 3$,

(ii) $v_\ell \in S$ if $y = 1$, $v_{\ell-1} \in S$ and $v_\ell \notin S$ if $y = 2$ and $v_\ell, v_{\ell-1} \notin S$ if $y = 3$, and

(iii) for every vertex v in $P - S$, $\deg_{P-S}(v) + w_v \le r$.

Here, we assume that $s_{x,y} = \infty$ if there exists no set fulfilling (i), (ii), and (iii).

Note that, given a weighted path $(P, w) \in \mathcal{P}_\ell$, we can compute its characteristic matrix in linear time by adapting Rules 3 to 5. We verified the following lemma using a computer program which enumerates the characteristic matrices of all weighted paths in $\mathcal{P}_\ell$, $\ell \le 7$.[(ii)] We remark that there are 11 distinct characteristic matrices arising from weighted paths on seven vertices.

Lemma 19. *For every weighted path $(P, w) \in \mathcal{P}_7$, there exists a weighted path $(P', w') \in \mathcal{P}_6$ such that $M(P, w) = M(P', w')$.*

Observe that, given a weighted path $(P, w) \in \mathcal{P}_7$, we can compute a shorter weighted path (P', w') such that $M(P, w) = M(P', w')$ in $\mathcal{O}(1)$ time. With this at hand, we can show that every maximal path can be replaced by a path on at most 6 vertices. For this, we need some additional notation. Let (G, w) be a graph and let $P = (v_1, \dots, v_\ell)$ be a path in G. We call $P - \{v_1, v_\ell\}$ the *inner path* of P and $V(P - \{v_1, v_\ell\})$ its *inner vertices*. Let $w^* \subseteq \mathbb{N}^{V(P)}$ be the weight vector obtained from w by restricting it to $V(P)$ and replacing the weights of v_1 and v_ℓ with $r - 1$, that is, $w^*_{v_1} = w^*_{v_\ell} = r - 1$ and $w^*_{v_i} = w_{v_i}$ for each inner vertex v_i.

Rule 6. Let $P = (v_1, \dots, v_7)$ be a (not necessarily maximal) path whose inner vertices have all degree two. Let $(P', w') \in \mathcal{P}_6$ be a weighted path such that $M(P', w') = M(P, w^*)$. Then replace the inner path of P with the inner path of P' and decrease k by $\operatorname{opt}(P, w^*) - \operatorname{opt}(P', w')$.

Lemma 20. *Rule 6 is correct.*

Proof: Let $\mathcal{I} = (G, k, r, w)$ be the instance of WBDD, which contains a path $P = (v_1, \dots, v_7)$ as described in Rule 6, and let $\mathcal{I}'$ be the instance of WBDD obtained from executing Rule 6. First, note that the existence of a weighted path $(P', w') \in \mathcal{P}_\ell$ in Rule 6 is guaranteed by Lemma 19. Suppose that $\mathcal{I}$ has a solution S. Let

$$x = \begin{cases} 1 & \text{if } v_1 \in S, \\ 2 & \text{if } v_1 \notin S, v_2 \in S, \\ 3 & \text{if } v_1, v_2 \notin S, \end{cases} \quad \text{and} \quad y = \begin{cases} 1 & \text{if } v_7 \in S, \\ 2 & \text{if } v_6 \in S, v_7 \notin S, \\ 3 & \text{if } v_6, v_7 \notin S. \end{cases}$$

Note that $|S \cap \{v_1, \dots, v_7\}| \ge \operatorname{opt}(P, w^*) + M(P, w^*)_{x,y}$ by Definition 18. In particular, it holds that $M(P, w^*)_{x,y} \ne \infty$. By the assumption that $M(P, w^*) = M(P', w')$, there exists a subset Q of vertices in $P' = (v'_1, \dots, v'_6)$ such that

(i) $v'_1 \in Q$ if $x = 1$, $v'_2 \in Q$ and $v'_1 \notin Q$ if $x = 2$, and $v'_1, v'_2 \notin Q$ if $x = 3$,

(ii) $v'_6 \in Q$ if $y = 1$, $v'_5 \in Q$ and $v'_6 \notin Q$ if $y = 2$, and $v'_5, v'_6 \notin Q$ if $y = 3$,

(iii) for every vertex v in $P' - Q$, $\deg_{P'-Q}(v) + w_v \le r$, and

[(ii)] The source code is made available at `https://git.tu-berlin.de/akt-public/bfvd-kernel`

(iv) $|Q| = \mathrm{opt}(P', w') + M(P, w^*)_{x,y}$.

We claim that $S' = (S \setminus \{v_2, \dots, v_6\}) \cup (Q \setminus \{v_1', v_6'\})$ is a solution of $\mathcal{I}'$. Since $|S \cap \{v_1, v_7\}| = |Q \cap \{v_1', v_6'\}|$, we have

$$\begin{aligned} |S'| &= |S| - |S \cap \{v_2, \dots, v_6\}| + |Q \setminus \{v_1', v_6'\}| \\ &= |S| - \big(|S \cap \{v_2, \dots, v_6\}| + |S \cap \{v_1, v_7\}|\big) + \big(|Q \setminus \{v_1', v_6'\}| + |Q \cap \{v_1', v_6'\}|\big) \\ &= |S| - |S \cap \{v_1, \dots, v_7\}| + |Q| \\ &\le k - \big(\mathrm{opt}(P, w^*) + M(P, w^*)_{x,y}\big) + \big(\mathrm{opt}(P', w') + M(P, w^*)_{x,y}\big) \\ &= k - \big(\mathrm{opt}(P, w^*) - \mathrm{opt}(P', w')\big). \end{aligned}$$

To verify that S' is a solution, it suffices to show that (1) $v_1 \in S'$ or S' contains at least $\deg(v_1) + w_{v_1} - r$ neighbors of v_1' in G' and (2) $v_7 \in S'$ or S' contains at least $\deg(v_7) + w_{v_7} - r$ neighbors of v_7 in G'. (Note that G' still contains the "original" endpoints v_1 and v_7 of P.) We only prove (1), since (2) can be shown analogously. If $x = 1$, then (1) clearly holds as $v_1 \in S$. If $x = 2$, then $v_1 \notin S$ and $v_2 \in S$; thus $S \setminus \{v_2, \dots, v_6\}$ contains at least $\deg(v_1) + w_{v_1} - r - 1$ neighbors of v_1, and as $S \setminus \{v_2, \dots, v_6\} \subseteq S'$ and $v_2' \in S'$, the set S' contains at least $\deg(v_1) + w_{v_1} - r$ neighbors of v_1. If $x = 3$, then $v_1, v_2 \notin S$; thus $S \setminus \{v_2, \dots, v_6\}$ contains at least $\deg(v_1) + w_{v_1} - r$ neighbors of v_1, and as $S \setminus \{v_2, \dots, v_6\} \subseteq S'$, the same holds for S'.

The other direction can be shown analogously because the proof of the forward direction does not rely on the fact that P' is shorter than P. □

Note that we can apply Rule 6 exhaustively in linear time since we have at most $|V(G)|$ applications of Rule 6, each of which take $\mathcal{O}(1)$ time to compute.

Proposition 21. *For constant r,* WBDD *has a kernel of size* $\mathcal{O}(\mathrm{fen})$.

Proof: We claim that after we apply Rules 3 to 6 exhaustively, we have an instance where the graph is of size $\mathcal{O}(\mathrm{fen})$. Note that Lemmas 16 and 20 establish the correctness of our rules. Moreover, we can apply these rule in linear time. Since Rules 3 to 5 delete all vertices of degree at most one, we have at most $2\mathrm{fen} - 2$ vertices of degree at least three by Lemma 17. Moreover, we have at most $3\mathrm{fen} - 3$ maximal paths and cycles whose internal vertices have degree two. By Rule 6, such a path or cycle is of length at most eleven. Since each degree-two vertex and each edge is contained in such a maximal path or cycle, the graph is of size $\mathcal{O}(\mathrm{fen})$. Finally, note that we need $\mathcal{O}(1)$ bits to encode each vertex weight. □

Removing weights. Towards showing that BDD has a kernel of size $\mathcal{O}(\mathrm{fen}^2)$, we use the following reduction rule to ensure that the weight of every vertex is at most $\mathcal{O}(\mathrm{fen})$.

Rule 7. If $w_v > 0$ for every vertex v, then decrease each w_v by one and decrease r by one.

Lemma 22. *Rule 7 is correct.*

Proof: By definition, for every vertex v, any solution must contain v or at least $\deg(v) + w_v - r = \deg(v) + (w_v - 1) - (r - 1)$ of its neighbors. □

Proposition 23. BDD *admits a kernel of size* $\mathcal{O}(\mathrm{fen}^2)$.

Proof: First, we show that after applying all our reduction rules, the weight of every vertex is at most $\mathcal{O}(\text{fen})$. Since Rule 7 has been applied exhaustively, there exists a vertex $v \in V$ with $w_v = 0$. If $r \geq \deg(v) + w_v = \deg(v)$, then Rule 4 was not exhaustively applied. Thus $r < \deg(v)$, which by Lemma 17 is in $\mathcal{O}(\text{fen})$. We have applied Rule 3, and hence for each vertex $v \in V(G)$, $w_v \leq r \in \mathcal{O}(\text{fen})$. An instance (G, k, r, w) of WBDD is equivalent to an instance (G', k, r) of BDD, where G' is a graph obtained by adding to w_v neighbors to every vertex v. Thus, we obtain a kernel of size $\mathcal{O}(\text{fen}^2)$. □

It is straightforward to adapt our algorithm to BFVD with $i \geq 2$:

Rule 8. If v is a vertex with $\deg(v) = 1$, then delete v.

Rule 8 is correct since a degree-one vertex is not part of any biclique when $i \geq 2$.

Rule 9. If $(v_1, v_2, v_3, v_4, v_5)$ is a path on five vertices with $\deg(v_i) = 2$ for each $i \in \{2, 3, 4\}$, then delete v_3.

Lemma 24. *Rule 9 is correct.*

Proof: It suffices to show that v_3 is not part of any biclique $K_{i,j}$ with $i \geq 2$. Suppose that v_3 is part of a biclique (S, T) with $|S| \geq 2$ and $|T| \geq 2$. Without loss of generality, assume that $v_3 \in S$. Then the only two neighbors of v_3, namely, v_2 and v_4 must be contained in T. Note, however, that $N(v_2) \cap N(v_4) = \{v_1, v_3\} \cap \{v_3, v_5\} = \{v_3\}$, implying that $|S| = 1$, a contradiction. □

One can apply Rules 8 and 9 exhaustively in linear time. By Lemma 17, we have the following:

Proposition 25. BFVD *admits a kernel of size* $\mathcal{O}(\text{fen})$ *for* $i \geq 2$.

Theorem 15 follows from Propositions 23 and 25.

7 Conclusion

In this work, we introduced the BICLIQUE FREE VERTEX DELETION (BFVD) problem and investigated its parameterized complexity with respect to structural parameters. We showed that BFVD is FPT for $d + k$, where d is the degeneracy and k is the solution size. This implies fixed-parameter tractability for the feedback vertex number fvn when $i \geq 2$. One natural question is whether the problem also admits a polynomial kernel for fvn. Recently, it was shown that all maximal bicliques can be enumerated efficiently on graphs of bounded weak closure (Koana et al., 2023), which is a superclass of degenerate graphs. Is BFVD also FPT when parameterized by the weak closure and the solution size?

References

P. Aboulker, É. Bonnet, E. J. Kim, and F. Sikora. Grundy coloring and friends, half-graphs, bicliques. *Algorithmica*, 85(1):1–28, 2023. URL https://doi.org/10.1007/s00453-022-01001-2.

N. Alon and S. Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. *Algorithmica*, 54(4):544–556, 2009. URL https://doi.org/10.1007/s00453-008-9204-0.

B. Balasundaram, S. S. Chandramouli, and S. Trukhanov. Approximation algorithms for finding and partitioning unit-disk graphs into co-k-plexes. *Optimization Letters*, 4(3):311–320, 2010. URL https://doi.org/10.1007/s11590-009-0146-5.

B. Balasundaram, S. Butenko, and I. V. Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011. URL https://doi.org/10.1287/opre.1100.0851.

M. Bentert, K. Heeger, and D. Knop. Length-bounded cuts: Proper interval graphs and structural parameters. *Journal of Computer and System Sciences*, 126:21–43, 2022.

N. Betzler, R. Bredereck, R. Niedermeier, and J. Uhlmann. On bounded-degree vertex deletion parameterized by treewidth. *Discrete Applied Mathematics*, 160(1):53–60, 2012. URL https://doi.org/10.1016/j.dam.2011.08.013.

H. L. Bodlaender and B. van Antwerpen-de Fluiter. Reduction algorithms for graphs of small treewidth. *Information and Computation*, 167(2):86–119, 2001. URL https://doi.org/10.1006/inco.2000.2958.

Z. Chen, M. R. Fellows, B. Fu, H. Jiang, Y. Liu, L. Wang, and B. Zhu. A linear kernel for co-path/cycle packing. In *Proceedings of the 6th International Conference on Algorithmic Aspects in Information and Management (AAIM 2010)*, pages 90–102, 2010. URL https://doi.org/10.1007/978-3-642-14355-7_10.

M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. URL https://doi.org/10.1007/978-3-319-21275-3.

A. Dessmark, K. Jansen, and A. Lingas. The maximum k-dependent and f-dependent set problem. In *Proceedings of the 4th International Symposium on Algorithms and Computation (ISAAC 1993)*, number 762 in LNCS, pages 88–97. Springer, 1993. URL https://doi.org/10.1007/3-540-57568-5_238.

R. Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer, 5th edition, 2017. URL https://doi.org/10.1007/978-3-662-53622-3.

R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Texts in Computer Science. Springer, 2013. URL https://doi.org/10.1007/978-1-4471-5559-1.

P. Dvořák and D. Knop. Parameterized complexity of length-bounded cuts and multicuts. *Algorithmica*, 80(12):3597–3617, 2018. URL https://doi.org/10.1007/s00453-018-0408-7.

E. Eiben, M. Kumar, A. E. Mouawad, F. Panolan, and S. Siebertz. Lossy kernels for connected dominating set on sparse graphs. *SIAM Journal on Discrete Mathematics*, 33(3):1743–1771, 2019. URL https://doi.org/10.1137/18M1172508.

D. Eppstein. Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, 51(4):207–211, 1994. URL https://doi.org/10.1016/0020-0190(94)90121-X.

L. Epstein, A. Levin, and G. J. Woeginger. The (weighted) metric dimension of graphs: Hard and easy cases. *Algorithmica*, 72(4):1130–1171, 2015. URL https://doi.org/10.1007/s00453-014-9896-2.

G. Fabianski, M. Pilipczuk, S. Siebertz, and S. Torunczyk. Progressive algorithms for domination and independence. In *Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019)*, pages 27:1–27:16, 2019. URL `https://doi.org/10.4230/LIPIcs.STACS.2019.27`.

M. R. Fellows, J. Guo, H. Moser, and R. Niedermeier. A generalization of nemhauser and trotter's local optimization theorem. *Journal of Computer and System Sciences*, 77(6):1141–1158, 2011. URL `https://doi.org/10.1016/j.jcss.2010.12.001`.

R. Ganian, F. Klute, and S. Ordyniak. On structural parameterizations of the bounded-degree vertex deletion problem. *Algorithmica*, 83(1):297–336, 2021. URL `https://doi.org/10.1007/s00453-020-00758-8`.

P. A. Golovach and Y. Villanger. Parameterized complexity for domination problems on degenerate graphs. In *Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008)*, pages 195–205, 2008. URL `https://doi.org/10.1007/978-3-540-92248-3_18`.

G. Gutin, M. Jones, and M. Wahlstrom. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM Journal on Discrete Mathematics*, 30(4):2177–2205, 2016. URL `https://doi.org/10.1137/15M1034337`.

L. Kellerhals and T. Koana. Parameterized complexity of geodetic set. *Journal of Graph Algorithms and Applications*, 26(4):401–419, 2022. URL `https://doi.org/10.7155/jgaa.00601`.

T. Koana, C. Komusiewicz, A. Nichterlein, and F. Sommer. Covering many (or few) edges with k vertices in sparse graphs. In P. Berenbrink and B. Monmege, editors, *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*, volume 219 of *LIPIcs*, pages 42:1–42:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL `https://doi.org/10.4230/LIPIcs.STACS.2022.42`.

T. Koana, C. Komusiewicz, and F. Sommer. Computing dense and sparse subgraphs of weakly closed graphs. *Algorithmica*, 85(7):2156–2187, 2023. URL `https://doi.org/10.1007/s00453-022-01090-z`.

C. Komusiewicz, F. Hüffner, H. Moser, and R. Niedermeier. Isolation concepts for efficiently enumerating dense subgraphs. *Theoretical Computer Science*, 410(38-40):3640–3654, 2009. URL `https://doi.org/10.1016/j.tcs.2009.04.021`.

M. Lampis and M. Vasilakis. Structural parameterizations for two bounded degree problems revisited. In *Proceedings of the 31st European Symposium on Algorithms (ESA 2023)*, 2023. URL `https://doi.org/10.48550/arXiv.2304.14724`. To appear.

B. Lin. The parameterized complexity of the k-biclique problem. *Journal of the ACM*, 65(5):34:1–34:23, 2018. URL `https://doi.org/10.1145/3212622`.

D. Lokshtanov, A. E. Mouawad, F. Panolan, M. S. Ramanujan, and S. Saurabh. Reconfiguration on sparse graphs. *Journal of Computer and System Sciences*, 95:122–131, 2018. URL `https://doi.org/10.1016/j.jcss.2018.02.004`.

D. Marx and P. Wollan. Immersions in highly edge connected graphs. *SIAM Journal on Discrete Mathematics*, 28(1):503–520, 2014. URL https://doi.org/10.1137/130924056.

B. McClosky and I. V. Hicks. Combinatorial algorithms for the maximum k-plex problem. *Journal of Combinatorial Optimization*, 23(1):29–49, 2012. URL https://doi.org/10.1007/s10878-010-9338-2.

H. Moser, R. Niedermeier, and M. Sorge. Exact combinatorial algorithms and experiments for finding maximum k-plexes. *Journal of Combinatorial Optimization*, 24(3):347–373, 2012. URL https://doi.org/10.1007/s10878-011-9391-5.

J. Nešetřil and P. Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, 2006. URL https://doi.org/10.1016/j.ejc.2005.01.010.

R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. ISBN 9780198566076. URL https://doi.org/10.1093/ACPROF:OSO/9780198566076.001.0001.

N. Nishimura, P. Ragde, and D. M. Thilikos. Fast fixed-parameter tractable algorithms for nontrivial generalizations of vertex cover. *Discrete Applied Mathematics*, 152(1):229–245, 2005. URL https://doi.org/10.1016/j.dam.2005.02.029.

V. Raman, S. Saurabh, and S. Srihari. Parameterized algorithms for generalized domination. In *Proceedings of the 2nd International Conference on Combinatorial Optimization and Applications (COCOA 2008)*, pages 116–126, 2008. URL https://doi.org/10.1007/978-3-540-85097-7_11.

S. B. Seidman and B. L. Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6:139–154, 1978. URL https://doi.org/10.1080/0022250X.1978.9989883.

J. A. Telle and Y. Villanger. FPT algorithms for domination in sparse graphs and beyond. *Theoretical Computer Science*, 770:62–68, 2019. URL https://doi.org/10.1016/j.tcs.2018.10.030.