Exploring the Kernelization Borders for Hitting Cycles

Akanksha Agrawal
Institute of Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary
akanksha.agrawal.2029@gmail.com

Pallavi Jain[1]
Institute of Mathematical Sciences, HBNI, Chennai, India
pallavij@imsc.res.in

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

Pranabendu Misra
University of Bergen, Bergen, Norway
Pranabendu.Misra@uib.no

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

Abstract

A generalization of classical cycle hitting problems, called conflict version of the problem, is defined as follows. An input is undirected graphs G and H on the same vertex set, and a positive integer k, and the objective is to decide whether there exists a vertex subset $X \subseteq V(G)$ such that it intersects all desired "cycles" (all cycles or all odd cycles or all even cycles) and X is an independent set in H. In this paper we study the conflict version of classical Feedback Vertex Set, Odd Cycle Transversal and Even Cycle Transversal problems, from the view point of kernelization complexity. In particular, we obtain the following results, when the conflict graph H belongs to the family of d-degenerate graphs.

1. CF-FVS admits a $\mathcal{O}(k^{\mathcal{O}(d)})$ kernel.
2. CF-ECT admits a $\mathcal{O}(k^{\mathcal{O}(d^2)})$ kernel.
3. CF-OCT does not admit polynomial kernel (even when H is 1-degenrate), unless $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$.

For our kernelization algorithms we exploit ideas developed for designing polynomial kernels for classical cycle hitting set problems, as well as, devise new reduction rules that exploit degeneracy crucially. Thus, in a broader sense, these kernelization algorithms generalize the known results for classical cycle hitting set problems (take H to be the edgeless graphs).

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

Keywords and phrases Parameterized Complexity, Kernelization, Conflict-free problems, Feedback Vertex Set, Even Cycle Transversal, Odd Cycle Transversal

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

[1] SERB-NPDF, India [PDF/2016/003508]

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:32
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Reducing the input data, in polynomial time, without altering the answer is one of the popular ways in dealing with intractable problems in practice. While such polynomial time heuristics can not solve NP-hard problems exactly, they work well on input instances arising in real-life. It is a challenging task to assess the effectiveness of such heuristics theoretically. Parameterized complexity, via kernelization, provides a natural way to quantify the performance of such algorithms. In parameterized complexity each problem instance comes with a parameter k and the parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm, called a *kernelization* algorithm, that reduces the input instance down to an instance with size bounded by a polynomial $p(k)$ in k, while preserving the answer. The reduced instance is called a $p(k)$ kernel for the problem.

The quest for designing polynomial kernels for "hitting cycles" in undirected graphs has played significant role in advancing the field of polynomial time pre-processing – kernelization. Hitting all cycles, odd cycles and even cycles correspond to well studied problems of Feedback Vertex Set (FVS), Odd Cycle Transversal (OCT) and Even Cycle Transversal (ECT), respectively. Alternatively, FVS, OCT and ECT correspond to deleting vertices such that the resulting graph is a forest, a bipartite graph and an odd cactus graph, respectively. All these problems, FVS, OCT, and ECT, have been extensively studied in parameterized algorithms and kernelization. The earliest known FPT algorithms for FVS go back to the late 80's and the early 90's [5, 13] and used the seminal Graph Minor Theory of Robertson and Seymour. On the other hand the parameterized complexity of OCT was open for long time. Only, in 2003, Reed et al. [27] gave a $3^k n^{\mathcal{O}(1)}$ time algorithm for OCT. This is also the paper which introduced the *method of iterative compression* to the field of parameterized complexity. However, the existence of polynomial kernel, for FVS and OCT were open questions for long time. For FVS, Burrage et al. [8] resolved the question in the affirmative by designing a kernel of size $\mathcal{O}(k^{11})$. Later, Bodlaender [6] reduced the kernel size to $\mathcal{O}(k^3)$, and finally Thomassé [28] designed a kernel of size $\mathcal{O}(k^2)$. The kernel of Thomassé [28] is best possible under a well known complexity theory hypothesis. It is important to emphasize that [28] popularized the method of *expansion lemma*, one of the most prominent approach in designing polynomial kernels. While, the kernelization complexity of FVS was settled in 2006, it took another 6 years and a completely new methodology to design polynomial kernel for OCT. Kratsch and Wahlström [19] resolved the question of existence of polynomial kernel for OCT by designing a randomized kernel of size $\mathcal{O}(k^{4.5})$ using matroid theory.[2] As a counterpart to OCT, Misra et al. [23] studied ECT and designed an $\mathcal{O}(k^3)$ kernel.

Fruitful and productive research on FVS and OCT have led to the study of several variants and generalizations of FVS and OCT. Some of these admit polynomial kernels and for some one can show that none can exist, unless some unlikely collapse happens in complexity theory. In this paper we study the following generlization of FVS, OCT and ECT, from the view-point of kernelization complexity.

Conflict Free Feedback Vertex Set (CF-FVS) **Parameter:** k
Input: An undirected graph G, a conflict graph H on vertex set $V(G)$ and a non-negative integer k.
Question: Does there exist $S \subseteq V(G)$, such that $|S| \leq k$, $G - S$ is a forest and $H[S]$ is edgeless?

[2] This foundational paper has been awarded the Nerode Prize for 2018.

One can similarly define Conflict Free Odd Cycle Transversal (CF-OCT) and Conflict-Free Even Cycle Transversal (CF-ECT).

Our Motivation. On the outset, a natural thought is "*why does one care*" about such an esoteric (or obscure) problem. We thought *exactly the same* in the beginning, till we realized the modeling power the problem provides and the rich set of questions one can ask. In the course of this paragraph we will try to explain this. First observe that, if one wants to model "independent" version of these problems (where the solution is suppose to be an independent set), then one takes conflict graph to be same as the input graph. An astute reader will figure out that the problem as stated above is W[1]-hard – a simple reduction from Multicolor Independent Set with each color class being modeled as cycle and the conflict graph being the input graph. Thus, a natural question is: *when does the problem become* FPT? To state the question formally, let $\mathcal{F}$ and $\mathcal{G}$ be two families of graphs. Then, $(\mathcal{G}, \mathcal{F})$-CF-FVS is same problem as CF-FVS, but the input graph G and the conflict graph H are restricted to belong to $\mathcal{G}$ and $\mathcal{H}$, respectively. It immediately brings several questions: (a) for which pairs of families the problem is FPT; (b) can we obtain some kind of dichotomy results; and (c) what could we say about the kernelization complexity of the problem. We believe that answering these questions for basic problems such as FVS, OCT, and Dominating Set will extend both the tractability as well as intractability tools in parameterized complexity and led to some fruitful and rewarding research. It is worth to note that initially we were inspired to define these problems by similar problems in computational geometry. See related results for more on this.

Our Results and Methods. A graph G is called *d-degenerate* if every subgraph of G has a vertex of degree at most d. For a fixed positive integer d, let $\mathcal{D}_d$ denote the set of graphs of *degeneracy* at most d. In this paper we study the $(\star, \mathcal{D}_d)$-CF-FVS ($\mathcal{D}_d$-CF-FVS) problem. The symbol $\star$ denotes that the input graph G is arbitrary. One can similarly define $\mathcal{D}_d$-CF-ECT. We also study, CF-OCT for a very restricted family of conflict graphs, a family of disjoint union of paths of length at most three and two star graphs. We denote this family as $\mathcal{P}^{\star\star}_{\leq 3}$ and this variant of CF-OCT as $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT. Starting point of our research is the recent study of Jain et al. [17], who studied conflict-free graph modification problems in the realm of parameterized complexity. As a part of their study they gave FPT algorithms for $\mathcal{D}_d$-CF-FVS, $\mathcal{D}_d$-CF-OCT and $\mathcal{D}_d$-CF-ECT using the independence covering families [20]. Their results also imply similar FPT algorithm when the conflict graph belongs to nowhere dense graphs. In this paper we focus on the kernelization complexity of $\mathcal{D}_d$-CF-FVS, $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT and $\mathcal{D}_d$-CF-ECT and obtain the following results.

1. $\mathcal{D}_d$-CF-FVS admits a $\mathcal{O}(k^{\mathcal{O}(d)})$ kernel.
2. $\mathcal{D}_d$-CF-ECT admits a $\mathcal{O}(k^{\mathcal{O}(d^2)})$ kernel.
3. $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT does not admit polynomial kernel, unless $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$.

Note that $\mathcal{D}_0$ denotes edgeless graphs and hence $\mathcal{D}_0$-CF-FVS, $\mathcal{D}_0$-CF-ECT and $\mathcal{D}_0$-CF-OCT are essentially FVS, ECT and OCT, respectively. Thus, any polynomial kernel for $\mathcal{D}_d$-CF-FVS, $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT, and $\mathcal{D}_d$-CF-ECT must generalize the known kernels for these problems. We remark that the above results imply that CF-FVS and CF-ECT admit polynomial kernels when the conflict graph belong to several well studied graph families, such as planar graphs, graphs of bounded degree, graphs of bounded treewidth, graphs excluding some fixed graph as a minor, a topological minor and graphs of bounded expansion etc. (all these graphs classes have bounded degeneracy).

***Strategy for* CF-FVS *and* $\mathcal{D}_d$-CF-ECT.** Our kernelization algorithm for CF-FVS consists of the following two steps. The first step of our kernelization algorithm is a structural decomposition of the input graph G. This does not depend on the conflict graph H. In this phase of the algorithm, given an instance (G, H, k) of CF-FVS we obtain an equivalent instance (G', H', k') of CF-FVS such that:

- The minimum degree of G' is at least 2.
- The number of vertices of degree at least 3 in G' is upper bounded by $\mathcal{O}(k^3)$. Let $V_{\geq 3}$ denote the set of vertices of degree at least 3 in G'.
- The number of maximal degree 2 paths in G' is upper bounded by $\mathcal{O}(k^3)$. That is, $G' - V_{\geq 3}$ consists of $\mathcal{O}(k^3)$ connected components where each component is a path.

We obtain this structural decomposition using reduction rules inspired by the quadratic kernel for FVS [28]. As stated earlier, this step can be performed for any graph H. Thus the problem reduces to designing reduction rules that bound the number of vertices of degree 2 in the reduced graph. Note that we can not do this for any arbitrary graph H as the problem is W[1]-hard. Once the decomposition is obtained we can not use the known *reduction rules* for FVS. This is for a simple reason that in G' the only vertices that are not bounded have degree exactly 2 in G'. On the other hand for FVS we can do simple "short-circuit" of degree 2 vertices (remove the vertex and add an edge between its two neighbors) and assume that there is no vertices of degree two in the graph. So our actual contributions start here. The second step of our kernelization algorithm bounds the degree two vertices in the graph G'. Here we must use the properties of the graph H. We propose new reduction rules for bounding degree two vertices, when H belongs to the family of d-degenerate graphs. Towards this we use the notion of d-degeneracy sequence, which is an ordering of the vertices in H such that any vertex can have at most d forward neighbors. This is used in designing a marking scheme for the degree two vertices. Broadly speaking our marking scheme associates a set with every vertex v. Here, set consists of " paths and cycles of G' on which the forward neighbors of v are". Two vertices are called similar if their associated sets are same. We show that if some vertex is not marked then we can safely contract this vertex to one of its neighbors. We then upper bound the degree two vertices by $\mathcal{O}(k^{\mathcal{O}(d)} d^{\mathcal{O}(d)})$, and thus obtain a kernel of this size for $\mathcal{D}_d$-CF-FVS.

The kernelization algorithm for $\mathcal{D}_d$-CF-ECT starts similar to the one for $\mathcal{D}_d$-CF-FVS, but we need to do significantly more work in this case to bound the vertices in simple structure we get after our decomposition. In particular, rather than degree two paths now we get chains of odd cycles. We first apply reduction rules, similar to the one to bound degree two paths in the case of $\mathcal{D}_d$-CF-FVS, to bound the length of these cycles. Then, we give a much more complicated marking schemes to show that if the chain is long then a cycle in this chain is irrelevant and hence can be "omitted". Also, note that, we can give a simple reduction from $\mathcal{D}_d$-CF-FVS to $\mathcal{D}_d$-CF-ECT by sub-dividing each edge and making sure that none of these are selected in solution. Towards this given an instance (G, H, k) of $\mathcal{D}_d$-CF-FVS, we generate an instance $(G', H', k+1)$ of $\mathcal{D}_d$-CF-ECT as follows. Initially, we have $V(G') = V(H') = V(G) \cup \{a, b_1, \cdots, b_{k+2}\}$. Now, for each edge $e_i \in E(G)$, add a vertex w_i to $V(G')$ and $V(H')$. Let x_i, y_i be end points of $e_i \in E(G)$. For each $e_i \in E(G)$, add edges $x_i w_i$ and $y_i w_i$ to $E(G')$. Also, add two multiple edges ab_i, for all $i \in [k+2]$. Edge set of H' is defined as $E(H') = E(H) \cup \{aw_i \mid w_i \in V(H')\}$. Since, we add $(k+2)$-even flowers in G' centred at a, a belongs to any solution of size $k+1$. Hence, the new vertices do not go to the solution. In the short version of the paper we only give key ideas in the polynomial kernel for $\mathcal{D}_d$-CF-FVS and have moved the complete section containing the kernel for $\mathcal{D}_d$-CF-ECT to the Appendix.

***Strategy for* CF-OCT.** The kernelization lower bound is obtained by the method of cross-composition [7]. We first define a conflict version of the s-t-Cut problem, where H belongs to $\mathcal{P}^{\star\star}_{\leq 3}$. Then, we show that the problem is NP-hard and cross composes to itself. Finally, we give a parameter preserving reduction from the problem to $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT, and obtain the desired kernel lower bound.

Related Work. In the past, the conflict free versions of some classical problems have been studied, e.g. for Shortest Path [18], Maximum Flow [24, 25], Knapsack [26], Bin Packing [14], Scheduling [15], Maximum Matching and Minimum Weight Spanning Tree [11, 10]. It is interesting to note that some of these problems are NP-hard even when their non-conflicting version is polynomial time solvable. The study of conflict free problems has also been recently initiated in computational geometry motivated by various applications (see [1, 2, 3]).

2 Preliminaries

Throughout the paper, we follow the following notions. Let G be a graph, $V(G)$ and $E(G)$ denote the vertex set and the edge set of graph G, respectively. Let n and m denote the number of vertices and the number of edges of G, respectively. Let G be a graph and $X \subseteq V(G)$, then $G[X]$ is the graph induced on X and $G - X$ is graph G induced on $V(G) \setminus X$. Let Δ denotes the maximum degree of graph G. We use $N_G(v)$ to denote the neighborhood of v in G and $N_G[v]$ to denote $N_G(v) \cup \{v\}$. Let E' be subset of edges of graph G, by $G[E']$ we mean the graph with the vertex set $V(G)$ and the edge set E'. Let $X \subseteq E(G)$, then $G - X$ is a graph with the vertex set $V(G)$ and the edge set $E(G) \setminus X$. Let Y be a set of edges on vertex set $V(G)$, then $G \cup Y$ is graph with the vertex set $V(G)$ and the edge set $E(G) \cup Y$. Degree of a vertex v in graph G is denoted by $deg_G(v)$. For an integer ℓ, we denote the set $\{1, 2, \ldots, \ell\}$ by $[\ell]$. A *path* $P = \{v_1, \ldots, v_n\}$ is an ordered collection of vertices such that there is an edge between every consecutive vertices in P and v_1, v_n are *endpoints* of P. For a path P by $V(P)$ we denote set of vertices in P and by $E(P)$ we denote set of edges in P. A *cycle* $C = \{v_1, \ldots, v_n\}$ is a path with an edge $v_1 v_n$. We define a *maximal degree two induced path in* G as an induced path of maximal length such that all vertices in path are of *degree exactly two in* G. An *isolated cycle* in graph G is defined as an induced cycle whose all the vertices are of degree exactly two in G. Let G' and G be graphs, $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$, then we say that G' is a *subgraph* of G. A graph is *connected* if any pair of vertices in the graph are joined by at least one path. A *cut vertex* of a graph G is a vertex such that $G[V(G) \setminus \{x\}]$ is not connected. A *maximal connected component* of a graph G a connected subgraphs of G with as many edges as possible. The subscript in the notations will be omitted if it is clear from the context.

A graph G has *degeneracy* d if every subgraph of G has a vertex of degree at most d. An ordering of vertices $\sigma : V(G) \to \{1, \cdots, n\}$ is is called a *d-degeneracy sequence* of graph G, if every vertex v has at most d neighbors u with $\sigma(u) > \sigma(v)$. A graph G is d-degenerate if and only if it has a d-degeneracy sequence. For a vertex v in d-degenerate graph G, the neighbors of v which comes *after (before)* v in d-degeneracy sequence are called *forward (backward) neighbors* of v in the graph G. Given a d-degenerate graph, we can find d-degeneracy sequence in linear time [21].

A *conflict graph*, H of graph G is a graph such that $V(H) = V(G)$ but edge sets might be different. A conflict free parameterization problem is denoted by (G, H, k) where H is a conflict graph and k is a parameter.

Two instances of a parameterized problem Π are called equivalent if $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi$. A *Reduction Rule* for a parameterized problem Π is a polynomial time algorithm which takes an instance (I, k) of Π and returns an instance (I', k') of Π. If (I, k) and (I', k') are equivalent then we say that reduction rule is *safe* or *correct*.

3 Tools: Polynomial Kernelization Algorithms

In this section, we give the tools and techniques which will be used in designing kernelization algorithms for CF-FVS and CF-ECT, when the conflict graph belongs to the family of d-degenerate graphs. Let σ be a d-degeneracy sequence of H which can be obtained in the polynomial time. In the kernelization algorithms, we will work with this fixed ordering. Forward and Backward neighbors of a vertex v are also defined with respect to ordering σ. If $\sigma(u) < \sigma(v)$, then u is a backward neighbor of v and v is a forward neighbor of u. By $N_H^f(v)$ ($N_H^b(v)$) we denote the set of forward (backward) neighbors of the vertex v in H.

Now, we give the technique which bounds the number of vertices in a d-degenerate graph H that share forward neighbors. Algorithm 1 describes this technique. We first define the notion of q-reducible set.

▶ **Definition 1.** For $q \in [d]$, let $q = 1$, $n_q = kd + 1$, otherwise $n_q = kn_{q-1} + kd + 1$. A set $X \subseteq V(H)$ of vertices is *q-reducible*, if for every set $U \subseteq X$ such that vertices in U share $d - q + 1$ forward neighbors, in particular $|\bigcap_{v \in U} N_H^f(v)| = d - q + 1$, we have that $|U| \leq n_q$.

Algorithm 1 Algo1(H, X)

Input: d-degenerate graph H, $X \subseteq V(H)$
Output: $X' \subseteq X$
For $q \in [d]$, let $n_q = kd + 1$, when $q = 1$, $n_q = kn_{q-1} + kd + 1$, otherwise.
while $q \leq d$ **do**
    while X is not q-reducible **do**
        Find $U \subseteq X$ of size $n_q + 1$, such that $|\bigcap_{v \in U} N_H^f(v)| = d - q + 1$.
        Let v be an arbitrary vertex in U.
        $X = X \setminus \{v\}$
    end while
    $q = q + 1$
end while
while $|X| > n_{d+1}$ **do**
    Let u be an arbitrary vertex in U.
    $X = X \setminus \{u\}$
end while
return $X' = X$

▶ **Observation 1.** *Let H be a d-degenerate graph and S be an independent set of H of size at most k. Then, for any set $U \subseteq V(H)$ such that for each vertex $u \in U$, $N_H^b(u) \cap S \neq \emptyset$, we have that $|U| \leq kd$.*

The proof of above observation follows from the fact that any vertex can have at most d forward neighbors.

▶ **Claim 1.** *Let d-degenerate graph H, $X \subseteq V(H)$ be the input and X' be the output of Algorithm 1. Let $S \subseteq V(H)$ be an independent set in H of size at most k and $v \in (X \setminus X') \cap S$. Then there exist a vertex $v' \in X'$ such that $(S \setminus \{v\}) \cup \{v'\}$ is also an independent set in H.*

Proof. We consider cases when X is d-reducible and when X is not d-reducible.
Case 1: X is not d-reducible.
We use induction on q to prove the lemma.

Base Step: $q = 1, n_1 = kd + 1$.
If X is not q-reducible, then the algorithm finds $U \subseteq X$ of size $kd+1$, such that $|\bigcap_{v \in U} N_H^f(v)| = d$, and deletes an arbitrary vertex $v \in U$ from X. If S contains v, then none of the forward neighbors of v are in S. This implies that none of the vertices in U have their forward neighbors in S. By Observation 1, at most kd vertices in U have backward neighbors in S. Therefore, there exists a vertex $v' \in U \setminus \{v\}$ such that neither any of its forward neighbor nor any of its backward neighbor is in S. Hence, $(S \setminus \{v\}) \cup \{v'\}$ is also an independent set in H of size at most k.

Induction Hypothesis: Let us assume that claim holds for $q \leq j - 1$.

Induction Step: $q = j, n_j = kn_{j-1} + kd + 1$.
If X is not j-reducible, then the algorithm finds $U \subseteq X$ of size n_j+1, such that $|\bigcap_{v \in U} N_H^f(v)| = d - j + 1$ and deletes an arbitrary vertex $v \in U$ from X. If S contains v, then none of the forward neighbors of v are in S. This implies that $d - j + 1$ common forward neighbors of vertices in U are not in S. Since, X is $j - 1$- reducible, therefore, if there exist a subset $U' \subseteq U$ such that vertices in U' have a vertex in S as a common forward neighbor, then this implies that $|\bigcap_{v \in U'} N_H^f(v)| \geq d - j + 2$ and by the induction hypothesis $|U'| \leq n_{j-1}$. Hence, at most kn_{j-1} vertices in U contains a vertex that have forward neighbors in S. By Observation 1, at most kd vertices in U have a backward neighbors in S. Therefore, there exists a vertex $v' \in U \setminus \{v\}$ such that neither any of its forward neighbor nor any of its backward neighbor is in S. Hence, $(S \setminus \{v\}) \cup \{v'\}$ is also an independent set in H of size at most k.

Case 2: X is d-reducible. When X is d-reducible and $|X| > n_{d+1}$, then the algorithm deletes an arbitrary vertex $v \in U$ from X. Let S contains v. Since, X is d- reducible, therefore, if there exist a subset $U' \subseteq U$ such that vertices in U' have a vertex in S as a common forward neighbor, then this implies that $|\bigcap_{v \in U'} N_H^f(v)| \geq 1$ and by correctness of the Case 1 $|U'| \leq n_d$. Hence, at most kn_d vertices in U contains a vertex that have forward neighbors in S. By Observation 1, at most kd vertices in U have backward neighbors in S. Therefore, there exists a vertex $v' \in U \setminus \{v\}$ such that neither any of its forward neighbor nor any of its backward neighbor is in S. Hence, $(S \setminus \{v\}) \cup \{v'\}$ is also an independent set in H of size at most k. ◀

▶ **Lemma 2.** *Algorithm 1 runs in polynomial time.*

Proof. Let σ be a d-degenerate sequence of the graph H. Using σ, for each $v \in X$, we can find $N_H^f(v)$ in the polynomial time. For $t \in [d]$, and $v \in V(H)$, we can find all the vertices in $V(H)$ that shares t forward neighbors with v, in polynomial time. Clearly, for a fixed q, Step 4 of Algorithm 1 can be performed in $\mathcal{O}(2^d)$ time. Since, $|X| \leq n$, this step can be performed in $\mathcal{O}(2^d n)$ time. Clearly, all the other steps of the algorithm can be performed in $\mathcal{O}(n)$ time. Hence, the running time of the algorithm is $n^{\mathcal{O}(1)}$. ◀

4 A Polynomial Kernel for $\mathcal{D}_d$-CF-FVS

In this section, we design a kernelization algorithm for $\mathcal{D}_d$-CF-FVS. To design a kernelization algorithm for $\mathcal{D}_d$-CF-FVS, we define another problem called $\mathcal{D}_d$-Disjoint-CF-FVS ($\mathcal{D}_d$-DCF-FVS, for short). We first define the problem $\mathcal{D}_d$-DCF-FVS formally, and then explain its uses in our kernelization algorithm.

$\mathcal{D}_d$-DISJOINT-CF-FVS ($\mathcal{D}_d$-DCF-FVS) **Parameter:** k
Input: An undirected graph G, a graph $H \in \mathcal{D}_d$ such that $V(G) = V(H)$, a subset $R \subseteq V(G)$, and a non-negative integer k.
Question: Is there a set $S \subseteq V(G) \setminus R$ of size at most k, such that $G - S$ does not have any cycle and S is an independent set in H?

Notice that $\mathcal{D}_d$-CF-FVS is a special case of $\mathcal{D}_d$-DCF-FVS, where $R = \emptyset$. Given an instance of $\mathcal{D}_d$-CF-FVS, the kernelization algorithm creates an instance of $\mathcal{D}_d$-DCF-FVS by setting $R = \emptyset$. Then it applies a kernelization algorithm for $\mathcal{D}_d$-DCF-FVS. Finally, the algorithm takes the instance returned by the kernelization algorithm for $\mathcal{D}_d$-DCF-FVS and generates an instance of $\mathcal{D}_d$-CF-FVS. Before moving forward, we note that the purpose of having set R is to be able to prohibit certain vertices to belong to a solution. This is particularly useful in maintaining the independent set property of the solution, when applying reduction rules which remove vertices from the graph (with an intention of it being in a solution).

We first focus on designing a kernelization algorithm for $\mathcal{D}_d$-DCF-FVS, and then give a polynomial time linear parameter preserving reduction from $\mathcal{D}_d$-DCF-FVS to $\mathcal{D}_d$-CF-FVS. If the kernelization algorithm for $\mathcal{D}_d$-DCF-FVS returns that (G, H, R, k) is a YES (NO) instance of $\mathcal{D}_d$-DCF-FVS, then conclude that (G, H, k) is a YES (NO) instance of $\mathcal{D}_d$-CF-FVS.In the following, we describe a kernelization algorithm for $\mathcal{D}_d$-DCF-FVS. Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS. The algorithm starts by applying the following simple reduction rules.

▶ **Reduction Rule 1.**
(a) *If $k \geq 0$ and G is acyclic, then return that (G, H, R, k) is a* YES *instance of* $\mathcal{D}_d$-DCF-FVS.
(b) *Return that (G, H, R, k) is a* NO *instance of* $\mathcal{D}_d$-DCF-FVS, *if one of the following conditions is satisfied:*
  (i) *$k \leq 0$ and G is not acyclic,*
  (ii) *G is not acyclic and $V(G) \subseteq R$, or*
  (iii) *There are more than k isolated cycles in G.*

▶ **Reduction Rule 2.**
(a) *Let v be a vertex of degree at most 1 in G. Then delete v from the graphs G, H and the set R.*
(b) *If there is an edge in G (H) with multiplicity more than 2 (more than 1), then reduce its multiplicity to 2 (1).*
(c) *If there is a vertex v with self loop in G. If $v \notin R$, delete v from the graphs G and H, and decrease k by one. Furthermore, add all the vertices in $N_H(v)$ to the set R, otherwise return that (G, H, R, k) is a* NO *instance of* $\mathcal{D}_d$-DCF-FVS.
(d) *If there are parallel edges between (distinct) vertices $u, v \in V(G)$ in G:*
  (i) *If $u, v \in R$, then return that (G, H, R, k) is a* NO *instance of* $\mathcal{D}_d$-DCF-FVS.
  (ii) *If $u \in R$ ($v \in R$), delete v (u) from the graphs G and H, and decrease k by one. Furthermore, add all the vertices in $N_H(v)$ ($N_H(u)$) to the set R.*

It is easy to see that the above reduction rules are correct, and can be applied in polynomial time. In the following, we define some notion and state some known results, which will be helpful in designing our next reduction rules.

▶ **Definition 2.** For a graph G, a vertex $v \in V(G)$, and an integer $t \in \mathbb{N}$, a *t-flower* at v is a set of t vertex disjoint cycles whose pairwise intersection is exactly $\{v\}$.

▶ **Proposition 1.** [9, 22, 28] *For a graph G, a vertex $v \in V(G)$ without a self-loop in G, and an integer k, the following conditions hold.*

(i) There is a polynomial time algorithm, which either outputs a $(k+1)$-flower at v, or it correctly concludes that no such $(k+1)$-flower exists. Moreover, if there is no $(k+1)$-flower at v, it outputs a set $X_v \subseteq V(G) \setminus \{v\}$ of size at most $2k$, such that X intersects every cycle passing through v in G.

(ii) If there is no $(k+1)$-flower at v in G and the degree of v is at least $4k + (k+2)2k$. Then using a polynomial time algorithm we can obtain a set $X_v \subseteq V(G) \setminus \{v\}$ and a set $\mathcal{C}_v$ of components of $G[V(G) \setminus (X_v \cup \{v\})]$, such that each component in $\mathcal{C}_v$ is a tree, v has exactly one neighbor in $C \in \mathcal{C}_v$, and there exist at least $k+2$ components in $\mathcal{C}_v$ corresponding to each vertex $x \in X_v$ such that these components are pairwise disjoint and vertices in X_v have an edge to each of their associated components.

▶ **Reduction Rule 3.** *Consider $v \in V(G)$, such that there is a $(k+1)$-flower at v in G. If $v \in R$, then return that (G, H, R, k) is a* NO *instance of $\mathcal{D}_d$-*DCF-FVS*. Otherwise, delete v from G, H and decrease k by one. Furthermore, add all the vertices in $N_H(v)$ to R.*

The correctness of above reduction rule follows from the fact that such a vertex must be part of every solution of size at most k. Moreover, the applicability of it in polynomial time follows from Proposition 1 (item (i)).

▶ **Reduction Rule 4.** *Let $v \in V(G)$, $X_v \subseteq V(G) \setminus \{v\}$, and $\mathcal{C}_v$ be the set of components which satisfy the conditions in Proposition 1(ii) (in G), then delete edges between v and the components of the set $\mathcal{C}_v$, and add parallel edges between v and every vertex $x \in X_v$ in G.*

The polynomial time applicability of Reduction Rule 4 follows from Proposition 1. And, in the following lemma, we prove the safeness of this reduction rule.

▶ **Lemma 3.** *Reduction Rule 4 is safe.*

Proof. Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS. Furthermore, let $v \in V(G)$, $X_v \subseteq V(G)$, and $\mathcal{C}_v$ be the tuple for which the conditions of Reduction Rule 4 are satisfied, and (G', H, R, k) be the instance resulting after application of the reduction rule. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-FVS if and only if (G', H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-FVS.

In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-FVS and S be one of its solution. We claim that S is also a solution of $\mathcal{D}_d$-DCF-FVS for (G', H, R, k). Suppose not, then $G' - S$ must contains a cycle as the conflict graphs in both the instances are the same. Observe that $G - \{v\}$ is identical to $G' - \{v\}$, and $G' - X_v$ is a subgraph of $G - X_v$, therefore, if either $v \in S$ or $X_v \subseteq S$, then S is a solution of $\mathcal{D}_d$-DCF-FVS for (G', H, R, k). Next, we assume that neither $v \notin S$, nor $X_v \nsubseteq S$. For $x \in X$, let $\mathcal{W}_x \subseteq \mathcal{C}_v$ be the set of components associated with x, which is obtained by the algorithm in Proposition 1(ii). Observe that, there are at least $k+2$ disjoint paths from v to each $x \in X_v$ passing through components in $\mathcal{W}_x$ in the graph G. Since S is of size at most k, there are at least two (distinct) connected components say C^1, C^2 in $\mathcal{W}_x$ such that v, x together with C^1, C^2 creates a cycle in $G - S$. This is a contradiction to S being a solution of $\mathcal{D}_d$-DCF-FVS for (G, H, R, k).

In the reverse direction, let (G', H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-FVS and S' be one of its solution. Observe that for each vertex $x \in X_v$, we have parallel edges between v and x in G', therefore either $v \in S'$ or $X_v \subseteq S'$. As observed before $G - \{v\}$ is identical to $G' - \{v\}$, therefore if $v \in S'$ then S' is also a solution of $\mathcal{D}_d$-DCF-FVS in (G, H, R, k). Next we assume that $X_v \subseteq S'$. Observe that edges incident to v and a vertex in some components in $\mathcal{C}_v$ are cut edges in $G - X_v$, by Proposition 1(ii), and hence they do not participate in any cycle in $G - X_v$. This concludes that S' is a solution of $\mathcal{D}_d$-DCF-FVS for (G, H, R, k). ◀

In the following state an easy observation, which follows from non-applicability of Reduction Rule 1 to 4.

▶ **Observation 4.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS, where none of Reduction Rule 1 to 4 apply. Then the degree of each vertex in G is bounded by $\mathcal{O}(k^2)$.*

To design our next reduction rule, we construct an auxilary graph $G^\star$. Intuitively speaking, $G^\star$ is obtained from G by shortcutting all degree two vertices. That is, vertex set of $G^\star$ comprises of all the vertices of degree at least three in 3. From now on, vertices of degree at least 3 (in G) will be refereed to as high degree vertices. For high degree vertex $v \in G$. For each $uv \in E(G)$, where u, v are high degree vertices, we add the edge uv in $G^\star$. Furthermore, for an induced maximal path P_{uv}, between u and v where all the internal vertices of P_{uv} are degree two vertices in G, we add the (multi) edge uv to $E(G^\star)$. Next, we will use the following result to bound the number of vertices and edges in $G^\star$.

▶ **Proposition 2.** [9] *A graph G with minimum degree at least 3, maximum degree Δ, and a feedback vertex set of size at most k has at most $(\Delta + 1)k$ vertices and $2\Delta k$ edges.*

The above result (together with the construction of $G^\star$) gives us the following (safe) reduction rule.

▶ **Reduction Rule 5.** *If $|V(G^\star)| \geq 4k^2 + 2k^2(k + 2)$ or $|E(G^\star)| \geq 8k^2 + 4k^2(k + 2)$, then return* No.

▶ **Lemma 5.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS, where none of the Reduction Rules 1 to 5 are applicable. Then we obtain the following bounds:*
- *The number of vertices of degree at least 3 in G is bounded by $\mathcal{O}(k^3)$.*
- *The number of maximal degree two induced paths in G is bounded by $\mathcal{O}(k^3)$.*

Having shown the above bounds, it remains to bound the number of degree two vertices in G. We start by applying the following simple reduction rule to eliminate vertices of degree two in G, which are also in R.

▶ **Reduction Rule 6.** *Let $v \in R$, $d_G(v) = 2$, and x, y be the neighbors of v in G. Delete v from the graphs G, H and the set R. Furthermore, add the edge xy in G.*

The correctness of this reduction rule follows from the fact that vertices in R can not be part of any solution and all the cycles passing through v also passes through its neighbors.

In the polynomial kernel for the Feedback Vertex Set problem (with no conflict constraints), we can short-circuit degree two vertices. But in our case, we cannot perform this operation, since we also need the solution to be an independent set in the conflict graph. Thus to reduced the number of degree two vertices in G, we exploit the properties of a d-degenerate graph. To this end, we use the tool that we developed in Section 3. This immediately gives us the following reduction rule.

▶ **Reduction Rule 7.** *Let P be a maximal degree two induced path in G. If $|V(P)| \geq n_{d+1} + 1$, apply Algorithm 1 with input $(H, V(P) \setminus R)$. Let $\widehat{V}(P)$ be the set returned by Algorithm 1. Let $v \in (V(P) \setminus R) \setminus \widehat{V}(P)$, and x, y be the neighbors of v in G. Delete v from the graphs G, H. Furthermore, add edge xy in G.*

▶ **Lemma 6.** *Reduction Rule 7 is safe.*

Proof. Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS and v be a vertex in a maximal degree two path P with neighbors x and y, with respect to which Reduction Rule 6 is applied. Furthermore, let (G', H', R, k) be the resulting instance after application of the reduction rule. We will show that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-FVS if and only if (G', H', R, k) is a YES instance of $\mathcal{D}_d$-DCF-FVS.

In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-FVS and S be one of its minimal solution. Consider the case when $v \notin S$. In this case, we claim that S is also a solution of $\mathcal{D}_d$-DCF-FVS for (G', H', R, k). Suppose not then either S is not an independent set in H' or $G' - S$ contains a cycle. Since, H' is an induced subgraph of H, we have that S' is also an independent set in H'. So we assume that $G' - S$ has a cycle, say C. If C does not contain the edge xy, then C is also a cycle in $G - S$. Therefore, we assume that C contains the edge xy. Bu then $(C \setminus \{xy\}) \cup \{xv, vy\}$ is a cycle in $G - S$. Next, we consider the case when $v \in S$. By Claim 1 we have a vertex $v' \in V(P) \setminus \{v\}$ such that $(S \setminus \{v\}) \cup \{v'\}$ is an independent set in H'. By using the fact that any cycle that passes through v also contains all vertices in P (together with the discussions above) imply that $(S \setminus \{v\}) \cup \{v'\}$ is a solution of $\mathcal{D}_d$-DCF-FVS for (G', H', R, k).

In the reverse direction, let (G', H', R, k) be a YES instance of $\mathcal{D}_d$-DCF-FVS and S' be one of its minimal solution. We claim that S' is also a solution of $\mathcal{D}_d$-DCF-FVS for (G, H, R, k). Suppose not, then either S is not an independent set in H or $G - S$ contains a cycle. Since, H' is an induced subgraph of H, we have that S' is also an independent set in H. Next, assume that there is a cycle C in $G - S$. The cycle C must contain v, otherwise, C is also a cycle in $G' - S'$. Since v is a degree two vertex in G, therefore any cycle that contians v, must also contain x and y. As observed before, $G - \{xv, vy\}$ is identical to $G' - \{xy\}$. But then, $(C \setminus \{xv, vy\}) \cup \{xy\}$ is a cycle in $G' - S'$, a contradiction. This concludes that S' is a solution of $\mathcal{D}_d$-DCF-FVS for (G, H, R, k). ◀

▶ **Lemma 7.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS, where none of the Reduction Rules 1 to 7 are applicable. Then the number of vertices in a degree two induced path in G is bounded by $\mathcal{O}(k^{\mathcal{O}(d)})$.*

▶ **Theorem 8.** *$\mathcal{D}_d$-DCF-FVS admits a kernel with $\mathcal{O}(k^{\mathcal{O}(d)})$ vertices.*

Proof. Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-FVS, where none of the Reduction Rules 1 to 7 are applicable. Then by Lemma 5, the number of vertices of degree at least 3 and the number of maximal degree two induced paths in G are bounded by $\mathcal{O}(k^3)$ and By Lemma 7, the number of vertices in a degree two induced path in G is bounded by $\mathcal{O}(k^{\mathcal{O}(d)})$. Hence, the number of vertices in G is bounded by $\mathcal{O}(k^{\mathcal{O}(d)})$. Since, each of the reduction rules can be applied in polynomial time and each of them either (correctly) declare that the given instance is a YES or NO instance or (safely) reduce the size of G. Therefore, the overall running time is polynomial in the input size. ◀

▶ **Lemma 9.** *There is a polynomial time parameter preserving reduction from $\mathcal{D}_d$-DCF-FVS to $\mathcal{D}_d$-CF-FVS.*

Proof. Given an instance (G, H, R, k) of $\mathcal{D}_d$-DCF-FVS, we generate an instance (G', H', k') of $\mathcal{D}_d$-CF-FVS as follows. We let the vertex set of $V(G')$ and $V(H')$ to be $V(G) \cup \{x\}$, where x is a new vertex. Now, we define the edge sets of G' and H'. Initially, $E(G') = E(G)$. Additionally, we add a self loop on x in G'. We let $E(H') = E(H - R) \cup \{xw \mid w \in R\}$. We set $k' = k + 1$. Clearly, this construction can be carried out in the running time linear in the size of the input instance. We claim that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-FVS if and only if $(G', H', k + 1)$ is a YES instance of $\mathcal{D}_d$-CF-FVS.

In the forward direction, let S be a solution to $\mathcal{D}_d$-DCF-FVS in (G, H, R, k). We claim that $S' = S \cup \{x\}$ is a solution to $\mathcal{D}_d$-CF-FVS in $(G', H', k+1)$. Since, $G' - \{x\}$ is identical to G, $G' - S'$ does not contain any cycle. Since, $S \cap R = \emptyset$, $S \cup \{v\}$ is an independent set in H'. This completes the proof in the forward direction. In the reverse direction, let (G', H', k') be a YES instance of $\mathcal{D}_d$-CF-FVS and S be one of its solution. Since there is a self loop at x in G, $x \in S$. We claim that $S' = S \setminus \{x\}$ is a solution to $\mathcal{D}_d$-DCF-FVS in (G, H, R, k). Clearly, $G' - \{x\}$ is identical to G, therefore, $G - S'$ does not contain any cycle. Since, $x \in S$, none of the vertices in R can belong to S. Since, $H - R$ same as $H - (R \cup \{x\})$, S' is an independent set in H' and $S' \cap R = \emptyset$, we have that S' is a solution to $\mathcal{D}_d$-DCF-FVS in (G, H, R, k). ◀

By Theorem 8 and Lemma 9, we obtain the following result.

▶ **Theorem 10.** $\mathcal{D}_d$-CF-FVS *admits a kernel with* $\mathcal{O}(k^{\mathcal{O}(d)})$ *vertices.*

5 Kernelization Complexity of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT

In this section, we show that CF-OCT does not admit a polynomial kernel when the conflict graph belongs to the family $\mathcal{P}^{\star\star}_{\leq 3}$. Let $\mathcal{P}_{\leq 3}$ denotes the family of disjoint union of paths of length at most three, and $\mathcal{P}^{\star}_{\leq 3}$ denotes the family of disjoint union of paths of length at most three and a star graph. We give parameter preserving reduction from $\mathcal{P}^{\star}_{\leq 3}$-CONFLICT FREE s-t CUT ($\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT) to $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT. $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT is formally defined as follows.

> $\mathcal{P}^{\star}_{\leq 3}$-CONFLICT FREE s-t CUT ($\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT) **Parameter:** k
> **Input:** An undirected graph G, a graph $H \in \mathcal{P}_{\leq 3}$ $(V(G) = V(H))$, two vertices s and t and an integer k
> **Question:** Is there a set $X \subseteq V$ such that X is a $s - t$ cut in G and $H[X]$ is edgeless?

We first prove that $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT is NP-hard. Then, we prove that $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT does not admit a polynomial compression, unless NP $\subseteq \frac{\text{coNP}}{\text{poly}}$ using the method of cross-composition. To show the NP-hardness of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT, we give a reduction from the well known NP-hard problem $(3, B2)$-SAT [4] to $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ CUT. $(3, B2)$-SAT is formally defined as follows.

> $(3, B2)$-SAT
> **Input:** An instance $(U, \mathcal{C})$, where U is the set of boolean variables and $\mathcal{C}$ is the set of clauses such that each clause has exactly three literals, and each literal occurs in exactly two clauses
> **Question:** Does there exist an assignment to variables such that each clause is satisfied?

5.1 NP-hardness of $\mathcal{P}_{\leq 3}$-CF-s-t CUT

In this section, we prove that $\mathcal{P}_{\leq 3}$-CF-s-t CUT is NP-hard. Given an instance $(U, \mathcal{C})$ of $(3, B_2)$-SAT, we construct an instance (G, H, s, t, k) of $\mathcal{P}_{\leq 3}$-CF-s-t CUT as follows. Let $|U| = n$ and $|\mathcal{C}| = m$. For each clause $C = (v_1, v_2, v_3) \in \mathcal{C}$, add vertices v_1^C, v_2^C, and v_3^C in $V(G)$ and $V(H)$. We also add $2n + 2$ new vertices s, t, a_i and b_i in $V(G)$ and $V(H)$, where $i \in [n]$. Corresponding to each clause $C = (v_1, v_2, v_3) \in \mathcal{C}$, we add a path $(s, v_1^C, v_2^C, v_3^C, t)$ in G. We also add paths (s, a_i, b_i, t), for all $i \in [n]$. Now we define edge set of H. Let $x_i \in U$. Add edges between a_i and vertices corresponding to positive literal of x_i and also between b_i and vertices corresponding to negative literal of x_i. We set $k = n + m$. Figure 1 describes

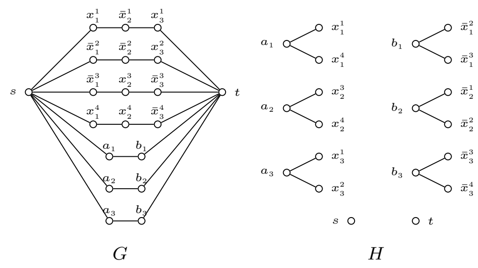

Figure 1 An illustration of construction of graph G and H in NP-hardness of $\mathcal{P}_{\leq 3}$-CF-s-t Cut for $\mathcal{C} = \{(x_1, \bar{x}_2, x_2), (\bar{x}_1, \bar{x}2, x_3), (\bar{x}_1, x_2, \bar{x}_3), (x_1, x_2, \bar{x}_3)\}$.

the construction of G and H. Clearly, this construction can be carried out in the polynomial time. In the following lemma, we prove that $\mathcal{C}$ is satisfiable if and only if (G, H) has a conflict free $s - t$ cut of size $n + m$.

▶ **Lemma 11.** *$(U, \mathcal{C})$ is a* YES *instance of* $(3, B_2)$-SAT *if and only if (G, H, s, t, k) is a* YES *instance of* $\mathcal{P}_{\leq 3}$-CF-s-t Cut.

Proof. In the forward direction, let $\mathcal{C}$ be satisfiable, and ϕ be a solution. Further, let S be the set of literals which are set to *true* in ϕ. Given S, we construct a solution S' of $\mathcal{P}_{\leq 3}$-CF-s-t Cut in (G, H) as follows. Let $v_i \in S$ and v_i belongs to the clauses C and C'. Add v_i^C and $v_i^{C'}$ to S'. Let $P_C = (s, v_1^C, v_2^C, v_3^C, t)$ be a path in G corresponding to the clause C. If more than one vertex from P_C belongs to S', delete all but one from S' arbitrarily. If variable corresponding to positive literal x_i belongs to S', add b_i to S', otherwise add a_i to S'. Since, there are $n + m$ disjoint paths between s and t and we select exactly one vertex from each path, $|S'| = n + m$. Since, $\mathcal{C}$ is satisfiable and for each path (s, a_i, b_i, t) either a_i or b_i belongs to S', S' is a $s - t$ cut of G. By the construction of S', it is also an independent set in H. This completes the proof in the forward direction.

In the reverse direction, let S be a solution to $\mathcal{P}_{\leq 3}$-CF-s-t Cut in (G, H, s, t, k). Given S, we construct a satisfyning assignment ϕ for the instance $(U, \mathcal{C})$ of $(3, B2)$-SAT as follows. Let v be a literal which occurs in the clauses C and C'. If $S \cap \{v^C, v^{C'}\} \neq \emptyset$, we assign 1 to v. Since, $H[S]$ is edgeless, if vertex corresponding to positive literal x_i belongs to the solution, b_i belongs to the solution and hence vertices corresponding to negative literal $\bar{x}_i$ do not belong to the solution. This implies that both the positive and negative literal corresponding to a variable are not set to one. If none of them are true, we assign 1 to x_i (or to $\bar{x}_i$). By the construction of G, ϕ is a satisfying assignment for $\mathcal{C}$. ◀

▶ **Theorem 12.** *$\mathcal{P}_{\leq 3}$-CF-s-t Cut is NP-hard.*

Proof. The proof follows from the construction of an instance of $\mathcal{P}_{\leq 3}$-CF-s-t Cut, Lemma 11 and NP-hardness of $(3, B_2)$-SAT. ◀

5.2 Lower bound for Kernel of $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ Cut

In this section, we prove that $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ Cut does not admit a polynomial compression unless $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$ which results into the fact that $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ Cut does not admit polynomial kernel as well. Towards this, we cross-compose $\mathcal{P}_{\leq 3}$-CF-s-t Cut into $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ Cut parameterized by k, the size of cut. Before going into the details, we define the notion of cross-composition.

▶ **Definition 3.** [7, 9] Let Σ be a finite set of alphabets. A *polynomial equivalence relation* is an equivalence relation $\mathcal{R}$ on $\Sigma^\star$ if there is an algorithm that given two strings $x, y \in \Sigma^\star$, decides whether $x \equiv_{\mathcal{R}} y$ in time polynomial in $|x| + |y|$. Moreover, the relation $\mathcal{R}$ restricted to the set $\Sigma^{\leq n}$ has at most $p(n)$ equivalence classes, where $p(\cdot)$ is some polynomial function.

▶ **Definition 4.** [7, 9] Let $L \subseteq \Sigma^\star$ be a language and $Q \subseteq \Sigma^\star \times \mathbb{N}$ be a parameterized language. We say that L *cross-composes* into Q if there exists a polynomial equivalence relation $\mathcal{R}$ and an algorithm $\mathcal{A}$ satisfying the following conditions. The algorithm $\mathcal{A}$ takes as input a sequence of strings $x_1, \cdots, x_t \in \Sigma^\star$ that are equivalent with respect to $\mathcal{R}$, runs in time polynomial in $\Sigma_{i=1}^{t}|x_i|$, and outputs one instance $(y, k) \in \Sigma^\star \times \mathbb{N}$ such that:

(i) $k \leq p(\max_{i\in[t]} |x_i| + \log t)$ for some polynomial $p(\cdot)$, and
(ii) $(y, k) \in Q$ if and only if there exists at least one index $i \in [t]$ such that $x_i \in L$.

Now, we state following known result which will be further used in this section.

▶ **Theorem 13.** [7, 9] *Let L be an NP-hard language that cross-composes into a parameterized language Q. Then, Q does not admit a polynomial compression, unless* $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$.

Next, we present a cross-composition of $\mathcal{P}_{\leq 3}$-CF-s-t CUT into $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ CUT parameterized by the solution size.

▶ **Lemma 14.** *There exists a cross-composition from $\mathcal{P}_{\leq 3}$-CF-s-t CUT into $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ CUT parameterized by the cut size.*

Proof. By choosing an appropriate polynomial equivalence relation $\mathcal{R}$, we may assume that we are given q instances $(G_i, H_i, s_i, t_i, k_i)_{i=1}^{q}$ of $\mathcal{P}_{\leq 3}$-CF-s-t CUT, where $V(G_i) = n$ and k_i is same for each $i \in [q]$. More precisely, equivalence relation $\mathcal{R}$ is defined as follows. We put all malformed instances into one equivalent class, while all the well-formed instances are partitioned with respect to the number of vertices in the graph and the integer k_i, where $i \in [q]$. Two well-formed instances are considered equivalent if number of the vertices in the graphs and integer k_i are same in both the instances. Clearly, the number of equivalence relation in $\Sigma^{\leq n}$ is bounded by $n^3 + 1$ and the equivalence of two relations can be tested in the polynomial time. Hence, $\mathcal{R}$ is a polynomial equivalence relation. The cross-composition algorithm works as follows. Given a set of malformed instances, returns some trivial no-instance of $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ CUT, while given a sequence of well-formed instances, it construct a parameterized instance $(G^\star, H^\star, s, t, k+1)$ of $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ CUT as follows. Let $x_i = (i-1)n+1$ and $y_i = x_i + n - 1$. Let $V(G_i) = V(H_i) = \{s_i, v_{x_i}, \cdots, v_{y_i}, t_i\}$. Now, we construct $G^\star$ and $H^\star$ as follows. $V(G^\star) = V(H^\star) = \cup_{i\in[q]}(V(G_i) \setminus \{s_i, t_i\}) \cup_{i\in[q-1]} \{w_i\} \cup \{a, s, t\}$ and $E(G^\star) = \cup_{i\in[q]}E(G_i - \{s_i, t_i\})$. If $s_1v \in E(G_1)$, add sv in $E(G^\star)$. Similarly, if $t_qv \in E(G_q)$, add tv in $E(G^\star)$. If an edge $t_iv \in E(G_i)$ or $s_{i+1}v \in E(G_{i+1})$, add an edge w_iv in $E(G^\star)$, for all $i \in [q-1]$. We also add edges sa and ta in $G^\star$. Now we define edge set of $H^\star$. $E(H^\star) = \cup_{i\in[q]}E(H_i - \{s_i, t_i\})$. We also add edge aw_i, for all $i \in [q-1]$. Since, paths are closed under vertex deletion, $H^\star$ belongs to the family $\mathcal{P}^\star_{\leq 3}$. We set parameter $k = k_1$. Figure 2 describes the construction of G and H. We claim that $(G^\star, H^\star, s, t, k+1)$ is a yes-instance of $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ CUT if and only if one of the input instance of $\mathcal{P}_{\leq 3}$-CF-s-t CUT has a conflict free $s - t$ cut of size k.

In the forward direction, let S be a solution to $\mathcal{P}^\star_{\leq 3}$-CF-$s$-$t$ CUT in $(G^\star, H^\star, s, t, k+1)$. Since, $a \in S$, none of w_i belongs to S, where $i \in [q-1]$. We claim that $S' = (S \setminus \{a\}) \cap V(G_i)$ is a solution to $\mathcal{P}_{\leq 3}$-CF-s-t CUT in $(G_i, H_i, s_i, t_i, k_i)$, for some $i \in [q]$. Suppose not, then there exists at least one path between each pair of vertex (s_i, t_i) in G_i, where $i \in [q]$. Let P_i be a path between s_i and t_i in G_i, where $i \in [q]$. Hence, path induced by the vertex

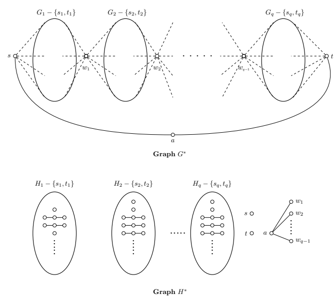

Figure 2 An illustration of construction of graph G and H in cross-composition from $\mathcal{P}_{\leq 3}$-CF-s-t Cut to $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut

set $\cup_{i\in[q]}(V(P_i) \setminus \{s_i,t_i\}) \cup_{i\in[q-1]} \{w_i\} \cup \{s,t\}$ yields a path between s and t in $G^{\star}$, a contradiction. Hence, there is some G_i such that S' is a $s-t$ cut of G_i. Since $H_i[S]$ is a induced subgraph of H, $H_i[S]$ is edgeless. This completes the proof in the forward direction.

In the reverse direction, let one of the input instance $(G_i, H_i, s_i, t_i, k_i), i \in [q]$ be a yes-instance of $\mathcal{P}_{\leq 3}$-CF-s-t Cut and S be one of its solution, i.e. $G_i \setminus S$ does not have a path from s_i to t_i. Clearly, by the construction of $G^{\star}$, $G^{\star} \setminus (S \cup \{a\})$ does not have a path from s to t. Since, a is not adjacent to any vertex $v \in V(H^{\star}) \cap V(H_i)$, where $i \in [q]$ and $uv \in E(H^{\star})$ if $uv \in E(H_i)$, where $i \in [q]$, $u \neq a$ and $v \neq a$, $S \cup \{a\}$ is an independent set in H. This completes the proof. ◀

▶ **Theorem 15.** $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut *does not admit a polynomial compression unless* $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$.

Proof. Since, $\mathcal{P}_{\leq 3}$-CF-s-t Cut is NP-hard, using Lemma 14 and Theorem 13, $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut, parameterized by the size of cut does not admit a polynomial compression unless $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$. ◀

▶ **Corollary 16.** $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut *does not admit a polynomial kernel.*

Proof. The proof follows from Theorem 15 and the fact that polynomial kernel is also a polynomial compression. ◀

5.3 Lower Bound for Kernel of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT

In this subsection, we prove the main result of this section. We show that there does not exist a polynomial kernel of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT. Towards this we give a parameter preserving reduction from $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut to $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT. Given an instance (G, H, s, t, k) of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut, we construct an instance $(G', H', k+1)$ of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT as follows. Initially, we have $V(G') = V(H') = V(G) \cup \{z, a, b\}$. Now, for each edge $e_i \in E(G)$, add a vertex w_i to $V(G')$ and $V(H')$. Now, we define the edge set of G'. Let x_i, y_i be end points of $e_i \in E(G)$. For each $e_i \in E(G)$, add edges x_iw_i and y_iw_i to $E(G')$. Also, add a self loop on z in G' and edges sa, ab and bt to $E(G')$. To construct the edge set of H', we set $E(H') = E(H - \{s,t\})$. Additionally, we add zs, zt, za, zt, and zw_i for each $w_i \in V(H')$ to $E(H')$. Figure 3 describes the construction of G and H. Clearly, H' belongs to $\mathcal{P}^{\star\star}_{3}$ and this construction can be carried out in the polynomial time. Now, we prove the equivalence between the instances (G, H, s, t, k) of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut and $(G', H', k+1)$ of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT in the following lemma.

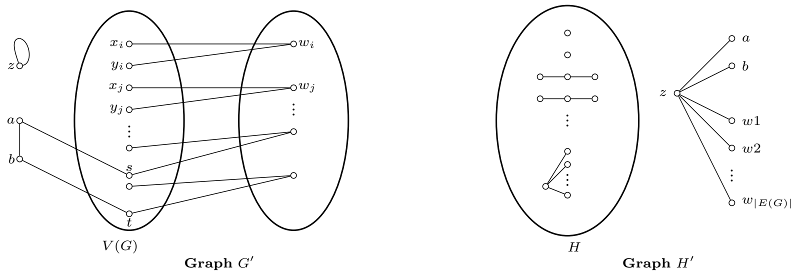

Figure 3 An illustration of construction of graph G and H in reduction from $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut to $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT.

▶ **Lemma 17.** *(G, H, s, t, k) is a yes-instance of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut if and only if $(G', H', k+1)$ is a yes-instance of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT.*

Proof. In the forward direction, let (G, H, s, t, k) be a yes-instance of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut and S be one of its solution. We claim that $S \cup \{z\}$ is a solution to $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT in $(G', H', k+1)$. In the graph G', since we subdivide each edge, all the paths from $s - t$ are of even length. Since, we subdivide each edge of G, $G' - \{a, b, z\}$ is a bipartite graph. Hence, an odd cycle in $G' - z$ consists of an $s - t$ path in $G' - \{a, b\}$ and edges sa, ab and bt. Clearly, by the construction of G', $(G' - \{a, b\}) \setminus S$ does not contain an $s - t$ path and hence $G' - z$ does not contain an odd cycle. Since, $H[S]$ is edgeless, $S \cup \{z\}$ is an independent set in H'. This completes the proof in the forward direction.

In the reverse direction, let S be a solution to $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT in $(G', H', k+1)$. Since, $z \in S$, therefore, $s, t, a, b, w_i \notin S$ for any $w_i \in V(H')$. We claim that $S' = S \setminus \{z\}$ is a solution to $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut in (G, H, s, t, k). Suppose not, then there exists a $s - t$ path $(s, x_1, x_2, \cdots, x_l, t)$ in $G \setminus S'$. Correspondingly, there exists a $s - t$ path $(s, w_1, x_1, w_2, x_2, \cdots, x_l, w_{l+1}, t)$ in G' of even length which results into an odd cycle $(s, w_1, x_1, w_2, x_2, \cdots, x_l, w_{l+1}, t, b, a)$ in $G' \setminus S$, a contradiction. This completes the proof. ◀

Now, we present the main result of this section in the following theorem.

▶ **Theorem 18.** *$\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT does not admit a polynomial kernel. unless $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$.*

Proof. Using the construction defined above, given an instance (G, H, s, t, k) of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut, we construct an instance $(G', H', k+1)$ of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT. Using Lemma 17, (G, H, s, t, k) is a yes-instance of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut if and only if $(G', H', k+1)$ is a yes-instance of $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT. We claim that $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT does not admit a polynomial kernel. Towards the contrary, suppose that $\mathcal{P}^{\star\star}_{\leq 3}$-CF-OCT admits polynomial kernel, then the instance (G, H, s, t, k) of $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut admits a polynomial compression, a contraction to the fact $\mathcal{P}^{\star}_{\leq 3}$-CF-$s$-$t$ Cut does not admit polynomial compression unless $\mathsf{NP} \subseteq \frac{\mathsf{coNP}}{\mathsf{poly}}$. ◀

References

1 Esther M. Arkin, Aritra Banik, Paz Carmi, Gui Citovsky, Matthew J. Katz, Joseph S. B. Mitchell, and Marina Simakov. Choice is hard. In *ISAAC*, pages 318–328, 2015.

2 Esther M. Arkin, Aritra Banik, Paz Carmi, Gui Citovsky, Matthew J. Katz, Joseph S. B. Mitchell, and Marina Simakov. Conflict-free covering. In *CCCG*, 2015.

3 Aritra Banik, Fahad Panolan, Venkatesh Raman, and Vibha Sahlot. Fréchet distance between a line and avatar point set. *FSTTCS*, pages 32:1–32:14, 2016.

4 Piotr Berman, Marek Karpinski, and Alex D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, (049), 2003.

5 Hans L. Bodlaender. On disjoint cycles. In Gunther Schmidt and Rudolf Berghammer, editors, *Proceedings on Graph–Theoretic Concepts in Computer Science (WG '91)*, volume 570 of *LNCS*, pages 230–238. Springer, 1992.

6 Hans L. Bodlaender. A cubic kernel for feedback vertex set. In *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, volume 4393 of *Lecture Notes in Computer Science*, pages 320–331. Springer, 2007.

7 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

8 Kevin Burrage, Vladimir Estivill-Castro, Michael R. Fellows, Michael A. Langston, Shev Mac, and Frances A. Rosamond. The undirected feedback vertex set problem has a poly(k) kernel. In *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, volume 4169 of *Lecture Notes in Computer Science*, pages 192–202. Springer, 2006.

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

10 Andreas Darmann, Ulrich Pferschy, and Joachim Schauer. Determining a minimum spanning tree with disjunctive constraints. In *ADT*, volume 5783 of *Lecture Notes in Computer Science*, pages 414–423. Springer, 2009.

11 Andreas Darmann, Ulrich Pferschy, Joachim Schauer, and Gerhard J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159(16):1726–1735, 2011.

12 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

13 Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness. In *Complexity Theory: Current Research*, pages 191–225. Cambridge University Press, 1992.

14 Leah Epstein, Lene M. Favrholdt, and Asaf Levin. Online variable-sized bin packing with conflicts. *Discrete Optimization*, 8(2):333–343, 2011.

15 Guy Even, Magnús M. Halldórsson, Lotem Kaplan, and Dana Ron. Scheduling with conflicts: online and offline algorithms. *J. Scheduling*, 12(2):199–224, 2009.

16 John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

17 Pallavi Jain, Lawqueen Kanesh, and Pranabendu Misra. Conflict free version of covering problems on graphs: Classical and parameterized. *To appear in CSR*, 2018.

18 Viggo Kann. *Polynomially bounded minimization problems which are hard to approximate*, pages 52–63. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

19 Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014.

20 Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Roohani Sharma, and Meirav Zehavi. Covering small independent sets and separators with applications to parameterized algorithms. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2785–2800, 2018.

21 David W. Matula and Leland L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM*, 30(3):417–427, 1983.

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

23 Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Parameterized algorithms for even cycle transversal. In *Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG*, volume 7551 of *Lecture Notes in Computer Science*. Springer.

24 Ulrich Pferschy and Joachim Schauer. The maximum flow problem with conflict and forcing conditions. In *INOC*, volume 6701 of *Lecture Notes in Computer Science*, pages 289–294. Springer, 2011.

25 Ulrich Pferschy and Joachim Schauer. The maximum flow problem with disjunctive constraints. *J. Comb. Optim.*, 26(1):109–119, 2013.

26 Ulrich Pferschy and Joachim Schauer. Approximation of knapsack problems with conflict and forcing graphs. *J. Comb. Optim.*, 33(4):1300–1323, 2017.

27 Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.

28 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010.

29 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.

A A Polynomial Kernel for CF-ECT

In this section, we design a kernelization algorithm for CF-ECT when the conflict graph belongs to the family of d-degenerate graphs, where $d \geq 1$. We call this variant of CF-ECT as $\mathcal{D}_d$-CF-ECT. In the following, we define the problem $\mathcal{D}_d$-CF-ECT formally.

$\mathcal{D}_d$-CF-ECT **Parameter:** k
Input: An undirected graph G, a d-degenerate graph H ($V(G) = V(H)$), where $d \geq 1$ and a non-negative integer k
Question: Is there a set $X \subseteq V(G)$ of size at most k, such that $G - S$ does not have any even cycle and S is an independent set in H?

We also define a variant of $\mathcal{D}_d$-CF-ECT, which we call $\mathcal{D}_d$-Disjoint-CF-ECT ($\mathcal{D}_d$-DCF-ECT) (to be defined shortly) as follows.

$\mathcal{D}_d$-Disjoint-CF-ECT ($\mathcal{D}_d$-DCF-ECT) **Parameter:** k
Input: An undirected graph G, a d-degenerate graph H ($V(G) = V(H)$), where $d \geq 1$, a subset $R \subseteq V(G)$ and a non-negative integer k.
Question: Is there a set $S \subseteq V(G) \setminus R$ of size at most k, such that $G - S$ does not have any even cycle and S is an independent set in H?

Given an instance of $\mathcal{D}_d$-CF-ECT, the kernelization algorithm first reduces to an instance of $\mathcal{D}_d$-DCF-ECT and applies the kernelization algorithm for $\mathcal{D}_d$-DCF-ECT. Finally the algorithm takes the reduced instance of $\mathcal{D}_d$-DCF-ECT and generates an instance of $\mathcal{D}_d$-CF-ECT. Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, purpose of the set R in $\mathcal{D}_d$-DCF-ECT is that the vertices in R are not to be picked in any solution of $\mathcal{D}_d$-CF-ECT for (G, H, k). Observe that given an instance (G, H, k) of $\mathcal{D}_d$-CF-ECT, we can construct an instance of $\mathcal{D}_d$-DCF-ECT by setting $R = \emptyset$. Firstly, we design a kernelization algorithm for $\mathcal{D}_d$-DCF-ECT and then give a polynomial time linear parameter preserving reduction from $\mathcal{D}_d$-DCF-ECT to $\mathcal{D}_d$-CF-ECT. If the kernelization algorithm for $\mathcal{D}_d$-DCF-ECT returns that (G, H, R, k) is a YES (NO) instance of $\mathcal{D}_d$-DCF-ECT, then conclude that (G, H, k) is a YES (NO) instance of $\mathcal{D}_d$-CF-ECT. Before going into the details of kernelization algorithm we define the notion of *cactus* and *odd cactus* graphs.

A block in a graph is a maximal connected subgraph without a cut vertex. A graph G is called a cactus graph if each edge in $E(G)$ is part of at most one cycle and each block in G is either an isolated vertex or an edge (*edge block*) or a cycle (*cycle block*). An odd cactus graph is a cactus graph without any even cycles. An *even cycle transversal* of a graph G is a set $S \subseteq V(G)$, such that $G - S$ does not contain any even cycle.

In the following we state some known results that will be used later to prove other lemmas.

▶ **Proposition 3.** [23] *Let G be a graph with at least two distinct cycles C, C' such that $|V(C) \cap V(C')| \geq 2$, then G has a cycle with even number of vertices.*

▶ **Proposition 4.** [23] *Let G be a graph and S be an even cycle transversal of G, then $G - S$ is an odd cactus graph.*

Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT. In the following, we give a kernelization algorithm for $\mathcal{D}_d$-DCF-ECT. Firstly, the algorithm applies some reduction rules to bound the maximum degree of a vertex in G. Then the algorithm uses an approximate solution S^* of ECT of size $\mathcal{O}(k)$ in G and exploit the properties of the graph $G - S^*$ and the conflict graph H to further reduce the instance. In the following sections, we state the reduction rules used by the algorithm. The kernelization algorithm applies all the reduction rules exhaustively, in the order in which they are stated.

A.1 Bounding Maximum Degree of G

Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT. In the following, we state some reduction rules that do not need the properties of graph H. We start with the following simple reduction rules.

▶ **Reduction Rule 8.**

(a) *If $k \geq 0$ and G is an odd cactus, then return that (G, H, R, k) is a* YES *instance of* $\mathcal{D}_d$-DCF-ECT.

(b) *Return that (G, H, R, k) is a* NO *instance of* $\mathcal{D}_d$-DCF-ECT, *if one of the following conditions is satisfied:*

  (i) *$k \leq 0$ and G is not an odd cactus,*

  (ii) *G is not an odd cactus and $V(G) \subseteq R$,*

  (iii) *There are more than k isolated cycles in G.*

▶ **Reduction Rule 9.**

(a) *Let v be a vertex of degree at most 1 in G. Then delete v from the graphs G, H and the set R.*

(b) *If there is an edge in G (H) with multiplicity more than 2 (more than 1) , then reduce its multiplicity to 2.*

(c) *If there is a vertex v with self loop in G, then delete self loop edge from vertex v in G.*

(d) *If there are parallel edges between (distinct) vertices $u, v \in V(G)$ in G:*

  (i) *If $u, v \in R$, then return that (G, H, R, k) is a* NO *instance of* $\mathcal{D}_d$-DCF-ECT.

  (ii) *If $u \in R$ $(v \in R)$, delete v (u) from the graphs G, H, decrease k by one. Furthermore, add all the vertices in $N_H(v)$ $(N_H(u))$ to the set R.*

▶ **Definition 5.** For a graph G and an even number $t \in \mathbb{N}$, a *t-even flower* at $v \in V(G)$ is a set of t vertex disjoint even cycles whose pairwise intersection is exactly v.

In the following we give some reduction rules to bound the maximum degree of a vertex in G. This requires the following proposition.

▶ **Proposition 5.** [23] *Given a graph G, a vertex $v \in V(G)$ and an integer k, there is a polynomial time algorithm such that, either it outputs a $(k+1)$-even flower at v, or it concludes that no such $(k+1)$-even flower exists and further outputs a set $S_v \subseteq V(G) \setminus \{v\}$ of cardinality $\mathcal{O}(k)$, such that $G - S_v$ has no even cycles.*

▶ **Reduction Rule 10.** *For a vertex $v \in V(G)$, apply algorithm in Proposition 5. Suppose that there exists a $(k+1)$-even flower at v in G. If $v \in R$, then return that (G, H, R, k) is a* NO *instance of* $\mathcal{D}_d$-DCF-FVS. *Otherwise, delete v from G and H and decrease k by one. Furthermore, add all the vertices in $N_H(v)$ to R.*

The correctness of above reduction rule follows from the fact that such a vertex must be part of every solution of size at most k.

Let S_v be the set obtained using Proposition 5 for a vertex $v \in V(G)$. Let $\mathcal{C}_v = C_1^v, \ldots, C_\ell^v$ be the set of connected components in $G[V(G) \setminus (S_v \cup \{v\})]$, that have a vertex adjacent to v in G.

We obtain the following observation by Proposition 3 and 4.

▶ **Observation 19.** *A vertex v can be adjacent to at most two vertices in each $C \in \mathcal{C}_v$ and each $C \in \mathcal{C}_v$ induces an odd cactus in G.*

▶ **Reduction Rule 11.** *Consider a vertex $v \in V(G)$. If there is a component $C_i^v \in \mathcal{C}_v$ such that there is no edge from C_i^v to S_v in G, in particular $|\bigcup_{u \in C_i^v}(N_G(u) \cap S_v)| = \emptyset$, then delete all vertices in C_i^v from G, H and the set R.*

The correctness of above reduction rule follows from the fact that vertices in such component can not be part of any even cycle in G.

▶ **Definition 6.** Let G be a bipartite graph with vertex bipartition (A, B). Let $X \subseteq A$ and $Y \subseteq B$, we say that X has $|X|$ many q-stars in Y if for every vertex $x \in X$ we can associate a subset $Z_x \subseteq N(x) \cap Y$ such that $|Z_x| = q$ and for any (distinct) $x, x' \in X$, we have that $Z_x \cap Z_{x'} = \emptyset$.

In the following we state a generalized version of *Expansion Lemma* in [29].

▶ **Proposition 6.** (q-*Expansion Lemma*)[29] Let $q \geq 1$ be a positive integer and G be a bipartite graph with bipartition (A, B) such that $|B| > q|A|$ and B does not contain isolated vertices. Then, there exists a polynomial time algorithm that outputs nonempty sets $X \subseteq A$ and $Y \subseteq B$ such that X has $|X|$ many q-stars in Y and $N(Y) \subseteq X$. In this case, we say that there is an q-expansion from X to Y.

For each vertex $v \in V(G)$ we define an auxiliary bipartite graph G_v with vertex bipartition (A_v, S_v), where $A_v = \{c \mid C \in \mathcal{C}_v\}$ and $E(G_v) = \{cy \mid xy \in E(G), x \in C, C \in \mathcal{C}_v, y \in S_v\}$. For a set $Y \subseteq A_v$, we define $E(v, Y) = \{vv' \mid vv' \in E(G), v' \in C, C \in \mathcal{C}_v, c \in Y\}$ (the set of edges in the graph G, between v and vertices in components corresponding to Y).

▶ **Reduction Rule 12.** *If for a vertex $v \in V(G)$, degree of v is at least $2(k+3)|S_v| + 2|S_v|$ in graph G, then apply algorithm in Proposition 6, to find sets $X_v \subseteq S_v$ and $Y_v \subseteq A_v$, such that there is a $(k+3)$-expansion from X_v to Y_v in the graph G_v. Then delete $E(v, Y_v)$ from G. Furthermore, add parallel edges between v and each vertex $x \in X_v$ in the graph G. Let (G', H, R, k) be the resulting instance.*

▶ **Lemma 20.** *Reduction rule 12 is safe.*

Proof. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G', H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT. In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-ECT, and let S be one of its solution. We claim that S is also a solution of $\mathcal{D}_d$-DCF-ECT for (G', H, R, k). Suppose not, then $G' - S$ contains an even cycle. Observe that $G - \{v\}$ is identical to $G' - \{v\}$, and $G' - X_v$ is a subgraph of $G - X_v$, therefore, if either $v \in S$ or $X_v \subseteq S$, then S is a solution of $\mathcal{D}_d$-DCF-ECT for (G', H, R, k). Next we assume that $v \notin S$ and $X_v \nsubseteq S$. Let $W_x \subseteq N(x) \cap Y_v$ be the set assigned to x by the expansion lemma and let $\mathcal{W}_x \subseteq \mathcal{C}_v$ be the set of connected components corresponding to set W_x. Observe that, there are $k+3$ disjoint paths from v to each $x \in X$ passing through each connected component in $\mathcal{W}_x$. Observe that, S can contain vertices from at most k connected components in $\mathcal{W}_x$. Therefore, there are at least three connected components say C^1, C^2, C^3 in $\mathcal{W}_x$ such that v, x together with C^1, C^2, C^3 creates an even cycle in $G - S$ by Proposition 3, a contradiction.

In the reverse direction, let (G', H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-ECT, and let S' be one of its solution. We claim that S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G', H, R, k). Suppose not, then $G' - S$ contains an even cycle C. As observed before, graph $G - \{v\}$ is identical to graph $G' - \{v\}$, therefore if $v \in S'$ then S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). Next we assume that $v \notin S'$. Observe that for each vertex $x \in X_v$, we have parallel edges between v and x in G', which creates an even cycle, therefore, $X_v \subseteq S'$.

Observe the only edges wich are not present in $G' - S'$ and present in $G - S'$ are edges in $E(y, Y_v)$, therefore C must contain an edge from the set $E(v, Y_v)$. By expansion lemma $N(Y_v) \subseteq X_v$, this implies that no vertex in any components corresponding to set $A_v \setminus Y_v$ have an edge to a vertex in $S_v \setminus X_v$ in G. By definition of S_v, every even cycle in G that passes from v also passes through a vertex in S_v. This implies that, any even cycle in G that passes through v and a vertex in components corresponding to the set Y_v must pass through X_v. Since $X_v \subseteq S'$, we obtain a contradiction. Therefore S' is a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). ◀

▶ **Lemma 21.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 12 are applicable. Then degree of each vertex in G is bounded by $\mathcal{O}(k^2)$.*

▶ **Proposition 7.** [23] *There is a factor 10-approximation algorithm for* ECT.

Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules from 8 to 12 are applicable. Next, the kernelization algorithm finds an approximate solution S^* of ECT in G, using Proposition 7. If $|S^*| > 10k$, then return that (G, H, R, k) is a NO instance of $\mathcal{D}_d$-DCF-ECT. The correctness follows from correctness of Proposition 7. From now onwards we assume that $|S^*| \leq 10k$. By Proposition 4, we know that $G - S^*$ is an odd cactus graph. In the following sections, the algorithm exploits the properties of the block decomposition of $G - S^*$ to further reduce the instance. Firstly, we define the notion of block decomposition and block graphs.

Given a graph $G^\star$, the *block decomposition* $\mathcal{D}$ of $G^\star$ is a set of all the blocks of the graph $G^\star$. The *block-cut vertex tree* T of G^* has vertices corresponding to each cut vertex in $G^\star$ and each block in $\mathcal{D}$. Let $u, v \in T$, where u corresponds to a cut vertex and v corresponds to a block B_v in $G^\star$. There is an edge uv in T, if $u \in B_v$. It is known that block-cut vertex tree of a graph is a tree [12]. A *pendant block* is a block which contains at most one cut vertex, equivalently a pendant block in $G - S^*$ is a block corresponding to a leaf vertex in T. A *high degree cut vertex* is a cut vertex with degree at least three in T, equivalently a high degree cut vertex is a cut vertex in $G - S^*$ which is contained in at least three blocks in $G - S^*$. A *low degree cut vertex* is a cut vertex with degree at most two in T, equivalently a low degree cut vertex is a cut vertex in $G - S^*$ which is contained in at most two blocks in $G - S^*$. A *high degree block* is a block which contains at least three cut vertices or contains a high degree cut vertex, equivalently a high degree block is a block in $G - S^*$ which corresponds to a vertex with degree at least three in T. A block B in a block chain $\mathcal{B}$ is called as *poor block* if both cut vertices of B are in R, $|E_{odd}(B)| = 1$, and $|E_{even}(B)| \geq 2$. A *dirty vertex* is a vertex that has a neighbor in S^*. A *dirty block* in $G - S^*$ is a block which contains a dirty vertex. Let P be a maximal degree two induced path in T such that blocks corresponding to P in $G - S^*$ are not dirty blocks, then the set of blocks corresponding to vertices in P in $G - S^*$ is called as *block chain*.

A.2 Bounding the number of high degree blocks and block chains

At this point, we have a graph G such that degree of all the vertices of G are bounded by $\mathcal{O}(k^2)$. In this section, we bound the number of pendant blocks, high degree blocks, high degree cut vertices, and number of block chains in $G - S^\star$.

▶ **Proposition 8.** [12, 16] *There is an algorithm, which given a graph G, in polynomial time outputs the block decomposition and block-cut vertex tree of G.*

The algorithm finds the block decomposition of $G - S^*$ and corresponding block graph T of graph $G - S^*$ using Proposition 8. Now, we first bound the number of pendant blocks.

▶ **Reduction Rule 13.** *If there is a pendant block B in $G - S^*$, which is not a dirty block, then delete all vertices except cut vertex in B, from the graphs G, H and the set R.*

The correctness of above reduction rule follows from the fact that vertices in such blocks do not participate in an even cycle in G. After applying the above reduction rule exhaustively, each leaf blocks in $G - S^*$ contains a vertex which has a neighbor in S^*. Using Lemma 21, we obtain the following result.

▶ **Lemma 22.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, when none of the Reduction Rules 8 to 13 are applicable. Then, the number of leaf blocks in $G - S^*$ and number of leaves in T are bounded by $\mathcal{O}(k^3)$.*

In the following we state a well known fact about trees.

▶ **Lemma 23.** [] *For a tree T with ℓ leaves, the number of degree at least three vertices in T is bounded by ℓ. Furthermore, the number of maximal degree two paths in T is bounded by ℓ.*

Using Lemmata 21, 22, and 23, we get the following result.

▶ **Lemma 24.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 13 are applicable. Then we obtain the following bounds.*

1. *The number of high degree blocks in $G - S^*$ is bounded by $\mathcal{O}(k^3)$.*
2. *The number of high degree cut vertices, dirty vertices and dirty blocks in $G - S^*$ is bounded by $\mathcal{O}(k^3)$.*
3. *The number of block chains in $G - S^*$ is bounded by $\mathcal{O}(k^3)$.*
4. *Let B is cycle block in $G - S^*$, then the number of maximal degree two induced paths in B, that do not contain a dirty vertex or cut vertex is bounded by $\mathcal{O}(k^3)$.*

After the exhaustive applications of the Reduction Rules 8 to 13, we are left with bounding number of vertices in a block and the number of blocks in a block chain.

A.3 Bounding number of vertices in a block

In this section, the kernelization algorithm exploits the properties of d-degenerate graph H to bound the vertices in a degree two induced path in G which results into bounding vertices in a block in $G - S^*$.

In the following, we state a reduction rule to bound vertices that belong to the set R in a degree two induced path in G.

▶ **Reduction Rule 14.** *If there exist a degree two induced path P in G such that P contains two vertices $u, v \in R$, then proceed as follows.*

- *If u, v are adjacent in G, let $x \neq v, y \neq u$ be neighbors of u, v in G respectively. Delete u, v from the graphs G, H and the set R. Furthermore, add edge xy in G.*
- *If u, v are not adjacent in G, let x_1, y_1 be neighbors of u and x_2, y_2 be neighbors of v in G. Delete u, v from the graphs G, H and the set R. Furthermore, add edges x_1y_1, x_2y_2 in G.*

Let (G', H', R', k) be the resulting instance.

▶ **Lemma 25.** *Reduction Rule 14 is safe.*

Proof. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G', H', R', k) is a YES instance of $\mathcal{D}_d$-DCF-ECT. In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S be one of its solution. We claim that S is also a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R', k). Suppose not, then either S is not an independent set in H' or $G'-S$ contains an even cycle. Since, H' is an induced subgraph of H, S is an independent set in H'. Let $G'-S$ contains an even cycle C. Observe that $G-\{u,v\}$ is identical to $G'-\{xy\}$ $(G'-\{x_1y_1, x_2y_2\})$. Therefore, if C does not contain edge xy (edges x_1y_1 or x_2y_2), then C is also an even cycle in $G-S$, a contradiction. Observe that any cycle in G' containing x_1y_1 also contains x_2y_2 and vice versa. Suppose that C contains edge xy (edges x_1y_1, x_2y_2), then $(C\setminus\{xy\})\cup\{xu, uv, vy\}$ $((C\setminus\{x_1y_1, x_2y_2\})\cup\{x_1u, uy_1, x_2v, vy_2\})$ is an even cycle in $G-S$, a contradiction. This concludes that S is a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R, k).

In the reverse direction, let (G', H', R', k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S' be one of its solution. We claim that S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). Suppose not, then either S' is not an independent set in H or $G-S'$ contains an even cycle. Since, H' is an induced subgraph of H, we have that S' is also an independent set in H. Let $G-S'$ contains an even cycle C. As observed before, $G-\{u,v\}$ is identical to $G'-\{xy\}$ $(G'-\{x_1y_1, x_2y_2\})$. Therefore if C does not contain u or v, then C is also an even cycle in $G'-S'$, a contradiction. Since P is a degree two induced path in G, therefor any cycle that passes trough one vertex (edge) of P passes through all (vertices) edges of P. Suppose that C contains vertices u, v then, $(C\setminus\{xu, uv, vy\})\cup\{xy\}$ $((C\setminus\{x_1u, uy_1, x_2v, vy_2\})\cup\{x_1y_1, x_2y_2\})$ is an even cycle in $G'-S'$, a contradiction. This concludes that S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). ◀

▶ **Lemma 26.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 14 are applicable, then for a degree two induced path P in G, $|V(P)\cap R|\leq 1$*

▶ **Reduction Rule 15.** *Let P be a maximal degree two induced path in G. If $|V(P)|\geq n_{d+1}+2$, apply Algorithm 1 with input $(V(P)\setminus R, H)$. Let $\widehat{V}(P)$ be the set returned by Algorithm 1. Let $u, v\in (V(P)\setminus R)\setminus \widehat{V}(P)$, then proceed as follows:*

- *If u, v are adjacent in G, let $x\neq v, y\neq u$ be neighbors of u, v in G respectively. Then delete u, v from the graphs G, H. Furthermore, add edge xy in G.*
- *If u, v are not adjacent in G, let x_1, y_1 be neighbors of u and x_2, y_2 be neighbors of v in G. Delete u, v from graph G, H. Add edges x_1y_1, x_2y_2 in G.*

Let (G', H', R, k) be the resulting instance.

▶ **Lemma 27.** *Reduction Rule 15 is safe.*

Proof. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G', H', R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT. In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S be one of its solution. Suppose that $S\cap\{u,v\}=\emptyset$, then we claim that S is also a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R, k). Suppose not, then either S is not an independent set in H' or $G'-S$ contains an even cycle. Since, H' is an induced subgraph of H, we have that S' is also an independent set in H'. Let $G'-S$ contains an even cycle C. Since P is a degree two induced path in G, any minimal solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k) contain at most one vertex from $V(P)$. Let $S\cap\{u,v\}=\emptyset$ and $G'-S$ contain an even cycle C. Observe that $G-\{u,v\}$ is identical to $G'-\{xy\}$ $(G'-\{x_1y_1, x_2y_2\})$. Therefore if C does not contain edge xy (edges x_1y_1 or x_2y_2), then C is also an even cycle in $G-S$, a contradiction. Observe that any cycle in G' containing x_1y_1 also contains x_2y_2 and vice versa. Suppose that C contains edge xy (edges x_1y_1,x_2y_2) then $(C\setminus\{xy\})\cup\{xu, uv, vy\}$ $((C\setminus\{x_1y_1, x_2y_2\})\cup\{x_1u, uy_1, x_2v, vy_2\})$ is an even cycle in $G-S$, a contradiction. This concludes that S is a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R, k). Now, suppose that S contains u or v, Without loss of generality, let $u\in S$, then by Claim 1 we have a vertex $u' \in \widehat{V}(P)$ such that $(S \setminus \{u\}) \cup \{u'\}$ is an independent set in H'. Since $\widehat{V}(P) \subset V(P)$, then by using the fact that any cycle that passes trough one vertex (edge) of P passes through all (vertices) edges of P, we have that $(S \setminus \{u\}) \cup \{u'\}$ is a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R, k).

Reverse direction arguments are similar to proof of Lemma 25.

◀

▶ **Lemma 28.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, when none of the Reduction Rules 8 to 15 are applicable. Then the number of vertices in a degree two induced path in G are bounded by $(d+1)k^{(d+1)}$.*

We obtain following result by Lemmata 24 and 28.

▶ **Lemma 29.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, when none of the Reduction Rules 8 to 15 are applicable. Then the number of vertices in a block in $G - S^*$ is bounded by $\mathcal{O}(dk^{\mathcal{O}(d)})$*

A.4 Bounding cut vertices in a block chain

In this section, the kernelization algorithm exploits the properties of d-degenerate graph H to bound the cut vertices in a block chain in $G - S^*$. Firstly, the kernelization algorithm applies reduction rules to bound the cut vertices that do not belong to R and then bounds the cut vertices that belong to the set R. Let B be a cycle block in $G - S^*$ with exactly two cut vertices a, b, then there are two disjoint paths from a to b in block B, one of even length and other of odd length. We define even (odd) length path as even-path (odd-path) in B. We define an *odd-even pair* (a, b) of a cycle block B as a pair of vertices such that vertex a belong to the odd-path and vertex b belong to the even-path in B. An odd-even pair (a, b) is *good* if $a, b \notin R$ and $ab \notin E(H)$. By $E_{odd}(B)(E_{even}(B))$, we denote edge set of odd-path (even-path) of B. Let $\mathcal{B}$ be a block chain in $G - S^*$, by $K(\mathcal{B})$ we denote cut vertices in $\mathcal{B}$ that are not in R. By $V(B)$ we denote vertices in a block B. By $V(\mathcal{B})$ we denote the set $\bigcup_{B \in \mathcal{B}} V(B)$.

In the following, we state a reduction rule to bound the number of cut vertices that do not belong to R.

▶ **Reduction Rule 16.** *Let $\mathcal{B}$ be a maximal block chain in $G - S^*$. If $K(\mathcal{B}) > n_{d+1}$, apply Algorithm 1, with input $(K(\mathcal{B}), H)$. Let $\widehat{K}(\mathcal{B})$ be the set returned by Algorithm 1 and $u \in K(\mathcal{B}) \setminus \widehat{K}(\mathcal{B})$. Then, add u to the set R to generate the reduced instance $(G, H, R' = R \cup \{u\}, k)$.*

▶ **Lemma 30.** *Reduction Rule 16 is safe.*

Proof. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G, H, R', k) is a YES instance of $\mathcal{D}_d$-DCF-ECT. In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S be one of its minimal solution. Since $\mathcal{B}$ is a block chain in $G - S^*$, any minimal solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k) contains at most one vertex from $K(\mathcal{B})$. Suppose that $u \notin S$. Since, $R' = R \cup \{u\}$, S is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R', k). Now, suppose that S contains u, then by Claim 1, we have a vertex $u' \in \widehat{K}(\mathcal{B})$ such that $S' = (S \setminus \{u\}) \cup \{u'\}$ is an independent set in H. Since $\mathcal{B}$ is a block chain, any cycle that contains a vertex in $\mathcal{B}$ passes through all cut vertices in $\mathcal{B}$ and hence S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). Hence, it is a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R', k).

In the reverse direction, since $R \subseteq R'$, any solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R', k) is also a solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k).

◀

▶ **Lemma 31.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT and none of the Reduction Rules 8 to 16 are applicable. Let $\mathcal{B}$ be a block chain in $G - S^*$. Then number of cut vertices in $\mathcal{B}$ that do not belong to the set R is bounded by $(d+1)k^{d+1}$.*

▶ **Lemma 32.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT and none of the Reduction Rules 8 to 16 are applicable. Let $\mathcal{B}$ be a block chain in $G - S^*$. Then the number of blocks in $\mathcal{B}$, which contains a cut vertex, that does not belong to the set R is bounded by $(d+1)k^{d+1}$.*

Now, we state a reduction rule to bound cut vertices with degree two in G, that belong to the set R in a block chain in $G - S^*$.

▶ **Reduction Rule 17.** *If there exist a block chain $\mathcal{B}$ in $G - S^*$ such that $\mathcal{B}$ contains two cut vertices $u, v \in R$ of degree exactly two in G, then proceed as follows.*

- *If u, v are adjacent in G, let $x \neq v, y \neq u$ be neighbors of u, v in G respectively. Then delete u, v from the graphs G, H and the set R. Furthermore, add edge xy in G.*
- *If u, v are not adjacent in G, let x_1, y_1 be neighbors of u and x_2, y_2 be neighbors of v in G. Then delete u, v from the graphs G, H and the set R. Furthermore, add edges x_1y_1 and x_2y_2 in G.*

Let (G', H', R', k) be the reduced instance.

▶ **Lemma 33.** *Reduction Rule 17 is safe.*

Proof. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G', H', R', k) is a YES instance of $\mathcal{D}_d$-DCF-ECT. In the forward direction, let (G, H, R, k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S be one of its minimal solution. We claim that S is also a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R', k). Suppose not, then either S is not an independent set in H' or $G' - S$ contains an even cycle. Since, H' is an induced subgraph of H, S is an independent set in H'. Let $G' - S$ contains an even cycle C. Observe that $G - \{u, v\}$ is identical to $G' - \{xv\}(G' - \{x_1y_1, x_2, y_2\})$. Therefore, if C does not contain edge xy (edges x_1y_1 or x_2y_2), then C is also an even cycle in $G - S$, a contradiction. Observe that any cycle that passes trough a vertex of a block chain passes through all of its cut vertices. This implies that any cycle in G' containing x_1y_1 also contains x_2y_2 and vice versa. Then $(C \setminus \{xy\}) \cup \{xu, uv, vy\}$ $((C \setminus \{x_1y_1, x_2y_2\}) \cup \{x_1u, uy_1, x_2v, vy_2\})$ is an even cycle in $G - S$, a contradiction. This concludes that S is a solution of $\mathcal{D}_d$-DCF-ECT for (G', H', R, k).

In the reverse direction, let (G', H', R', k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S' be one of its solution. We claim that S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). Suppose not, then either S' is not an independent set in H or $G - S'$ contains an even cycle. Since, H' is an induced subgraph of H, we have that S' is also an independent set in H. Let $G - S'$ contains an even cycle C. As observed before, $G - \{u, v\}$ is identical to $G' - \{xy\}$ $(G' - \{x_1y_1, x_2y_2\})$. Therefore if C does not contain u or v, then C is also an even cycle in $G' - S'$, a contradiction. As observed before any cycle that passes trough a vertex of a block chain passes through all of its cut vertices. Suppose that C contains vertices u, v then, $(C \setminus \{xu, uv, vy\}) \cup \{xy\}$ $((C \setminus \{x_1u, uy_1, x_2v, vy_2\}) \cup \{x_1y_1, x_2y_2\})$ is an even cycle in $G' - S'$, a contradiction. This concludes that S' is also a solution of $\mathcal{D}_d$-DCF-ECT for (G, H, R, k). ◀

▶ **Lemma 34.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 17 are applicable. Then there exist at most one cut vertex of degree two in G that belong to the set R, in a block chain in $G - S^*$.*

At this point, we are only left with bounding the cut vertices of degree at least 3 in G that belong to R in a block chain in $G - S^*$. Towards this, we bound the number of cycle blocks in a block chain that contains both of the cut vertices in the set R, which also results into bounding cut vertices in $G - S^*$ that belong to R and of degree at least 3 in G.

▶ **Lemma 35.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 17 are applicable. Then a vertex in G can be in at most $(d+1)k^{d+1}$ odd-even pairs in $G - S^*$.*

Proof. The number of vertices in a degree two induced path in G is bounded by $(d+1)k^{d+1}$ by Lemma 28. This implies that number of vertices in odd and even paths in a block are bounded by $(d+1)k^{d+1}$. Hence, the claim follows. ◀

▶ **Observation 36.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT. Let $\mathcal{B}$ be a block chain in $G - S^*$, with at least two cycle blocks. Let B be a block in $\mathcal{B}$ with both of its cut vertices in R. Then for any minimal solution S of (G, H, R, k) for $\mathcal{D}_d$-DCF-ECT, either $V(B) \cap S = \emptyset$, or $V(B) \cap S = \{a, b\}$ such that (a, b) is a good odd-even pair in B, or $V(B) \cap S = \{v\}$ such that B is a poor block, $v \in B$, and $v \notin R$.*

Let $\mathcal{B}$ be a block chain in $G - S^*$. By $\mathcal{B}_P$ we denote the set of all poor blocks in $\mathcal{B}$, which have both the cut vertices in R. By $W(\mathcal{B}_P)$ we denote the set of all vertices that do not belong to the set R in blocks in $\mathcal{B}_P$.

▶ **Reduction Rule 18.** *Let $\mathcal{B}$ be a block chain in $G - S^*$. If $|W(\mathcal{B}_P)| \geq k + 1$, then add all vertices of $W(\mathcal{B}_P)$ to R to generate a reduced instance (G, H, R', k).*

▶ **Lemma 37.** *Reduction Rule 18 is safe.*

Proof. We prove that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G, H, R', k) is a YES instance of $\mathcal{D}_d$-DCF-ECT. In the forward direction, let S be a minimal solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). Let $B_1, \cdots, B_\ell$ be poor blocks in $\mathcal{B}$, then we claim that if a vertex from even-path in B_i is in S for any $i \in [\ell]$, then $|S \cap B_j| = 1$ for all $j \in [\ell]$. Since, S is a minimal solution, there exists an even cycle C that contains $E_{even}(B_i)$. Since, $\mathcal{B}$ is a block chain, either $E_{odd}(B_j)$ or $E_{even}(B_j)$ is contained in C for $j \neq i$. If C contains $E_{odd}(B_j)$, then $(C \setminus (E_{odd}(B_j \cup E_{even}(B_i))) \cup (E_{even}(B_j \cup E_{odd}(B_i)))$ is an even cycle in $G - S$, a contradiction. If C contains $E_{even}(B_j)$, then $(C \setminus (E_{even}(B_j) \cup E_{even}(B_i)) \cup (E_{odd}(B_j) \cup E_{odd}(B_i))$ is an even cycle in $G - S$, a contradiction. Since, $|W(\mathcal{B}_P)| \geq k + 1$, no vertex of $W(\mathcal{B}_P)$ belongs to S. Hence, S is also a solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R', k). This completes the proof in the forward direction.

In the reverse direction, since $R \subseteq R'$, any solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R', k) is also a solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). ◀

▶ **Lemma 38.** *Let (G, H, R, k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 18 are applicable. Then the number of poor blocks in a block chain in $G - S^*$ is bounded by k.*

In the following, we state reduction rules to bound cycle blocks that contain both of the cut vertices in R and are not poor. Let $\mathcal{B}$ be a block chain in $G - S^*$. By $\mathcal{B}_R$ we denote the set of all cycle blocks in $\mathcal{B}$, which have both the cut vertices in R. By $O(\mathcal{B}_R)$ we denote the set of all good odd-even pairs in blocks in $\mathcal{B}_R$. For a set A of odd-even pairs by $V(A)$ we denote the set $\bigcup_{(a,b)\in A}\{a,b\}$.

▶ **Definition 7.** *Let $\eta = (d+1)k^{(d+1)}$. If $r = 1$, let $n_r = \eta(kd)+1$, otherwise, $n_r = kn_{r-1} + \eta(kd) + 1$. A set X of odd-even pairs is* r-reducible*, if for every set $U \subseteq X$ such that pairs in U have at least $2d - r + 1$ common forward neighbors, in particular $|\bigcap_{(a,b)\in U}(N_H^f(a) \cup N_H^f(b))| = 2d - r + 1$, $|U| \le n_r$.*

The following observation is due to Observation 1 and Lemma 28

▶ **Observation 39.** *Let (G, H, R, k) be a* YES *instance of* $\mathcal{D}_d$-DCF-ECT *and S be one of its solution. Let X be a set of good odd-even pairs such that for each pair $(a,b) \in X$, either a or b have a backward neighbor in S, in particular $(N_H^b(a) \cup N_H^b(b)) \cap S \neq \emptyset$, then $|X| \le \eta(kd)$, where $\eta = (d+1)k^{(d+1)}$.*

Algorithm 2 Algo2(H, X)

Input: d-degenerate graph H, $X \subseteq V(H) \times V(H)$
Output: $X' \subseteq X$

For $r \in [2d]$, let $n_r = \eta(kd)+1$, if $r = 1$, otherwise, $n_r = \eta n_{r-1} + \eta(kd) + 1$.
$q = 1$
while $r \le 2d$ **do**
    while X is not r-reducible **do**
        Find $U \subseteq X$ of size $n_r + 1$, such that $|\bigcap_{(a^*,b^*)\in U}(N_H^f(a^*) \cup N_H^f(b^*))| = 2d - r + 1$.
        Let (a,b) be an arbitrary pair in U.
        $X = X \setminus \{(a,b)\}$
    end while
    $q = q + 1$
end while
while $|X| > n_{2d+1}$ **do**
    Let (x,y) be an arbitrary pair in X.
    $X = X \setminus \{(x,y)\}$
end while
return $X' = X$

▶ **Claim 2.** *Let d-degenerate graph H, $X \subseteq V(H) \times V(H)$ be the input and X' be the output of Algorithm 2. Let $S \subseteq V(H)$ be an independent set in H of size at most k and $(a,b) \in (X \setminus X') \cap S$. Then there exist a pair $(a',b') \in X'$ such that $(S \setminus \{a,b\}) \cup \{a',b'\}$ is also an independent set in H of size at most k.*

Proof. We consider cases when X is $2d$-reducible and when X is not $2d$-reducible.
Case 1: X is not $2d$-reducible.
We use induction on r to prove the lemma.
Base Step: $r = 1, n_1 = \eta(kd) + 1$.

If X is not r-reducible, then the algorithm finds $U \subseteq X$ of size $\eta(kd) + 2$, such that $|\bigcap_{(a^*,b^*)\in U}(N_H^f(a^*) \cup N_H^f(b^*))| = 2d$, and delete an arbitrary pair $(a,b) \in U$ from X. If S contains a, b, then none of the forward neighbors of a, b are in S. This implies that none of the vertices in $V(U)$ have their forward neighbors in S. By Observation 39, at most $\eta(kd)$ pairs in U contains a vertex that have a backward neighbors in S. Therefore, there exists a pair $(a',b') \in U \setminus \{(a,b)\}$ such that vertices a', b' have neither any of its forward neighbor nor any of its backward neighbor in S. Hence, $(S \setminus \{a,b\}) \cup \{a',b'\}$ is also an independent set in H of size at most k.

Induction Hypothesis: Let us assume that claim holds for $r \leq j-1$.

Induction Step: $r = j, n_j = kn_{j-1} + \eta(kd) + 1$.

If X is not j-reducible, then algorithm finds $U \subseteq X$ of size $n_j + 1$, such that $|\bigcap_{(a^*,b^*)\in U}(N^f_H(a^*) \cup N^f_H(b^*))| = 2d-j+1$ and delete an arbitrary pair $(a,b) \in U$ from X. If S contains a,b, then none of the forward neighbors of a,b are in S. This implies that $2d-j+1$ common forward neighbors of vertices of pairs in U are not in S. Since, X is $(j-1)$-reducible, therefore, if there exist a subset $U' \subseteq U$ such that vertices in pairs of U' have a vertex in S as a common forward neighbor, then this implies that $|\bigcap_{(a^*,b^*)\in U'}(N^f_H(a^*) \cup N^f_H(b^*))| \geq 2d-j+2$ and by the induction hypothesis $|U'| \leq n_{j-1}$. Hence, at most kn_{j-1} pairs in U contains a vertex that have forward neighbors in S. By Observation 39, at most $\eta(kd)$ pairs in U contains a vertex that have a backward neighbors in S. Therefore, there exists a pair $(a',b') \in U \setminus \{(a,b)\}$ such that vertices a',b' have neither any of its forward neighbor nor any of its backward neighbor are in S. Hence, $(S \setminus \{a,b\}) \cup \{a',b'\}$ is also an independent set in H of size at most k.

Case 2: X is $2d$-reducible.

When X is $2d$-reducible and $|X| > n_{2d+1}$, then algorithm delete an arbitrary pair $(a,b) \in X$ from X. Let S contains a,b. Since, X is $2d$-reducible, therefore, if there exist a subset $U' \subseteq X$ such that vertices in pairs of U' have a vertex in S as a common forward neighbor, then this implies that $|\bigcap_{(a^*,b^*)\in U'}(N^f_H(a^*) \cup N^f_H(b^*))| \geq 1$ and by correctness of the Case 1, we have that $|U'| \leq n_{2d}$. Hence, at most kn_{2d} pairs in X contains a vertex that have forward neighbors in S. By Observation 39, at most $\eta(kd)$ pairs in X contains a vertex that have a backward neighbors in S. Therefore, there exists a pair $(a',b') \in X \setminus \{(a,b)\}$ such that vertices a',b' have neither any of its forward neighbor nor any of its backward neighbor in S. Hence, $(S \setminus \{a,b\}) \cup \{a',b'\}$ is also an independent set in H of size at most k. ◀

▶ **Lemma 40.** *Algorithm 2 runs in polynomial time.*

Proof. Let σ be a d-degenerate sequence of the graph H. Using σ, for each $v \in V(H)$, we can find $N^f_H(v)$ in the polynomial time. For $t \in [2d]$, and a pair of distinct vertices $u,v \in V(H)$, we can find all the pairs of vertices in $V(H)$ that shares t forward neighbor with u,v in polynomial time. Clearly, for a fixed r, Step 4 of Algorithm 1 can be performed in $\mathcal{O}(2^{2d})$ time. Since, $X \subseteq \binom{V(G)}{2}$, $|X| \leq n^2$, hence Step 4 can be performed in $\mathcal{O}(2^d n^2)$ time. Clearly, all the other steps of the algorithm can be performed in $\mathcal{O}(n)$ time. Hence, the running time of the algorithm is $n^{\mathcal{O}(1)}$. ◀

▶ **Reduction Rule 19.** *Let $\mathcal{B}$ be a block chain in $G - S^*$. If $|\mathcal{B}_R| > n_{2d+1}$, apply Algorithm 2 with input $(O(\mathcal{B}_R), H)$. Let $\widehat{O}(\mathcal{B}_R)$ be the set returned by Algorithm 2. Let $B_1 \in \mathcal{B} \setminus \mathcal{B}_T$ be a cycle block that do not contain any pair in $\widehat{O}(\mathcal{B}_R)$, and x_1, y_1 be the cut vertices of B_1. Since, $\mathcal{B}$ is a block chain, either x_1 or y_1 has at least three neighbors in $V(\mathcal{B})$. Without loss of generality, let $|N_G(y_1) \cap V(\mathcal{B})| \geq 3$, then proceed as follows:*

Case 1. *Let $|N_G(y_1) \cap V(\mathcal{B})| = 3$ and $w \in (N_G(y_1) \cap V(\mathcal{B})) \setminus V(B_1)$. Delete $V(B_1) \setminus \{x_1, y_1\}$ from the graphs G, H and the set R. Furthermore, add edges x_1w, x_1y_1 to G.*

Case 2. *Let $|N_G(y_1) \cap V(\mathcal{B})| = 4$ and $w_1, w_2 \in (N_G(y_1) \cap V(\mathcal{B})) \setminus V(B_1)$. Delete $V(B_1) \setminus \{x_1\}$ from the graphs G, H and the set R. Furthermore, add the edges x_1w_1 and x_1w_2 to G.*

▶ **Lemma 41.** *Reduction Rule 19 is safe.*

Proof. Let (G', H', R', k) be the reduced instance after the application of Reduction Rule 19. We claim that (G, H, R, k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if (G', H', R', k) is a YES instance of $\mathcal{D}_d$-DCF-ECT.

Case 1. Suppose that the instance (G', H', R', k) is generated by the application of Case 1. In the forward direction, let S be a minimal solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). If S is also a solution of $\mathcal{D}_d$-DCF-ECT in (G', H', R', k), then the claim holds. Otherwise, S is not a solution of $\mathcal{D}_d$-DCF-ECT in (G', H', R', k). This implies that either S is not an independent set in H' or $G' - S$ contains an even cycle. Since, H' is an induced subgraph of H, S is also an independent set in H'. Let C be an even cycle in $G' - S$. Suppose that $S \cap V(B_1) = \emptyset$. Since, $G - (V(B_1) \setminus \{x_1, y_1\})$ is identical to $G' - \{x_1w, x_1y_1\}$, C contains either edges $\{x_1y_1, y_1w\}$ or x_1w. If C contains $\{x_1y_1, y_1w\}$, then $(C \setminus \{x_1y_1\}) \cup E_{odd}(B_1)$ is an even cycle in $G - S$, a contradiction. If C contains x_1w, then $(C \setminus \{x_1w\}) \cup E_{even}(B_1) \cup \{y_1w\}$ is an even cycle in $G - S$, a contradiction to the fact that S is a solution of $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). Now, suppose that $S \cap V(B_1) \neq \emptyset$. Since, all the vertices of a poor block in $\mathcal{B} \setminus \mathcal{B}_T$ are red, B_1 is not a poor block. Using Observation 36, if $a \in S \cap V(B_1)$, then there exists a vertex $b \in S \cap V(B_1)$ such that (a, b) is a good odd-even pair in B_1. By claim 2 there exists a pair $(a', b') \in \widehat{O}(\mathcal{B}_R)$ such that $S' = (S \setminus \{a, b\}) \cup \{a', b'\}$ is an independent set in H'. Observe that if a cycle C passes through a vertex in $V(\mathcal{B})$, then either $C \cap E_{odd}(B) \neq \emptyset$ or $C \cap E_{even}(B) \neq \emptyset$, for any $B \in \mathcal{B}$. Therefore, S' is a solution of $\mathcal{D}_d$-DCF-ECT in (G', H', R', k).

In the reverse direction, let (G', H', R', k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S be one of its solution. If S is also a solution of $\mathcal{D}_d$-DCF-ECT in (G, H, R, k), then the claim holds. Otherwise, S is not a solution of $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). Since, H' is an induced subgraph of H, S is an independent set in H. Let C be an even cycle in $G - S$. Since, $G - (V(B_1) \setminus \{x_1, y_1\})$ is identical to $G' - \{x_1w, x_1y_1\}$, either C contains $E_{odd}(B_1)$ or $E_{even}(B_1)$. Also, $C \cap \{y_1w_1\} \neq \emptyset$. Suppose C contains $E_{odd}(B_1)$, then $(C \setminus E_{odd}(B_1)) \cup \{x_1y_1\}$ is an even cycle in $G' - S$, a contradiction. If C contains $E_{even}(B_1)$ then $(C \setminus (E_{even}(B_1) \cup \{y_1w_1\})) \cup \{x_1w\}$ is an even cycle in $G' - S$, a contradiction. Hence, S is also a solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k).

Case 2. Suppose that the instance (G', H', R', k) is generated by the application of Case 2. Let B_2 be a block which contains w_1 and w_2. Without loss of generality, let $y_1w_1 \in E_{odd}(B_2)$ and $y_1w_2 \in E_{even}(B_2)$. In the forward direction, let S be a minimal solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). If S is also a solution of $\mathcal{D}_d$-DCF-ECT in (G', H', R', k), then the claim holds. Otherwise, S is not a solution to $\mathcal{D}_d$-DCF-ECT in (G', H', R', k). This implies that either S is not an independent set in H' or $G' - S$ contains an even cycle. Since, H' is an induced subgraph of H, S is also an independent set in H'. Let C be an even cycle in $G' - S$. Suppose that $S \cap V(B_1) = \emptyset$. Since, $G - (V(B_1) \setminus \{x_1\})$ is identical to $G' - \{x_1w_1, x_1w_2\}$, C contains either x_1w_1 or x_1w_2. Then, $(C \setminus \{x_1w_i\}) \cup E_{even}(B_1) \cup \{y_1w_i\}$, where $i \in [2]$ is an even cycle in $G - S$, a contradiction. Now, suppose that $S \cap V(B_1) \neq \emptyset$. By claim 2 there exists a pair $(a', b') \in \widehat{O}(\mathcal{B}_R)$ such that $S' = (S \setminus \{a, b\}) \cup \{a', b'\}$ is an independent set in H'. Observe that if a cycle C passes through a vertex in $V(\mathcal{B})$, then either $C \cap E_{odd}(B) \neq \emptyset$ or $C \cap E_{even}(B) \neq \emptyset$, for any $B \in \mathcal{B}$. Therefore, S' is a solution of $\mathcal{D}_d$-DCF-ECT in (G', H', R', k).

In the reverse direction, let (G', H', R', k) be a YES instance of $\mathcal{D}_d$-DCF-ECT and S be one of its solution. If S is also a solution of $\mathcal{D}_d$-DCF-ECT in (G, H, R, k), then the claim holds. Otherwise, S is not a solution of $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). This implies that either S is not an independent set in H or $G - S$ contains an even cycle. Since, H' is an induced subgraph of H, S is an independent set in H. Let C be an even cycle in $G - S$.

Since, $G-(V(B_1)\setminus\{x_1\})$ is identical to $G'-\{x_1w_1,x_1w_2\}$, C contains either $E_{odd}(B_1)$ or $E_{even}(B_1)$. Suppose that $C\cap E_{odd}(B_i)\neq\emptyset$, for all $i\in[2]$, then $(C\setminus(E_{odd}(B_1)\cup E_{odd}(B_2)))\cup(E_{even}(B_2)\setminus\{y_1w_2\})\cup\{x_1w_2\}$ is an even cycle in $G'-S$, a contradiction. The similar argument follows when $C\cap E_{even}(B_i)\neq\emptyset$, for all $i\in[2]$. Suppose that C contains $E_{odd}(B_1)$ and $E_{even}(B_2)$, then $(C\setminus(E_{odd}(B_1)\cup E_{even}(B_2)))\cup(E_{odd}(B_2)\setminus\{y_1w_1\})\cup\{x_1w_1\})$ is an even cycle in $G'-S$, a contradiction. The similar argument follows when C contains $E_{even}(B_1)$ and $E_{odd}(B_2)$. Hence S is also a solution to $\mathcal{D}_d$-DCF-ECT in (G,H,R,k) ◀

▶ **Lemma 42.** *Let (G,H,R,k) be an instance of* $\mathcal{D}_d$-DCF-ECT*, where none of the Reduction Rules 8 to 19 are applicable. Then, the number of blocks in a block chain in $G-S^*$, which contains both cut vertices in R is bounded by $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$.*

▶ **Lemma 43.** *Let (G,H,R,k) be an instance of* $\mathcal{D}_d$-DCF-ECT*, where none of the Reduction Rules 8 to 19 are applicable. Then the number of blocks in a block chain in $G-S^*$ is bounded by $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$.*

Proof. The number of blocks in a block chain which contains a cut vertex, that does not belong to the set R is bounded by $(d+1)k^{d+1}$ by Lemma 32. Since, the number of blocks in a block chain in $G-S^*$, which contains both of the cut vertices in R is also bounded by $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$ by Lemma 42, the claim follows. ◀

▶ **Theorem 44.** $\mathcal{D}_d$-DCF-ECT *admits a kernel with $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$ vertices.*

Proof. Let (G,H,R,k) be an instance of $\mathcal{D}_d$-DCF-ECT, where none of the Reduction Rules 8 to 19 are applicable. Let $S^\star$ be an approximate solution of ECT for G. Let T be a block-cut vertex tree of $G-S^\star$. By Lemma 22, number of leaf blocks are bounded by $\mathcal{O}(k^3)$. By Lemma 24, number of high degree blocks, high degree cut vertices, number of degree two induced paths which neither contain a cut vertex nor a dirty vertex in a block, and the number of block chains are bounded by $\mathcal{O}(k^3)$. The algorithm further bounds number of vertices in a block in $G-S^\star$ by $\mathcal{O}(dk^{\mathcal{O}(d)})$ by Lemma 29. Now, we are only left with bounding cut vertices in a block chain. By Lemma 43, the number of blocks in a block chain are bounded by $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$. Hence, the number of cut vertices in a block chain are also bounded by $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$. Since, each of the rules can be applied in polynomial time and each of them either declare that the given instance is a YES or NO instance or reduce the size of the graph. Therefore, the overall running time is polynomial in the input size.

◀

▶ **Lemma 45.** *There is a parameter preserving reduction from* $\mathcal{D}_d$-DCF-ECT *to* $\mathcal{D}_d$-CF-ECT*.*

Proof. Given an instance (G,H,R,k) of $\mathcal{D}_d$-DCF-ECT, we generate an instance (G',H',k') of $\mathcal{D}_d$-CF-ECT as follows. Let $X=\{v,u_1,\cdots,u_{k+2}\}$. We define vertex set of $V(G')$ and $V(H')$ as $V(G)\cup X$. Now, we define edge sets of G' and H'. $E(G')=E(G)\cup\{vu_i,vu_i\mid i\in[k+2]\}$. $E(H')=E(H-R)\cup\{vw\mid w\in R\}$. We set $k'=k+1$. Clearly, this construction can be carried out in the polynomial time in the size of input instance. We claim that (G,H,R,k) is a YES instance of $\mathcal{D}_d$-DCF-ECT if and only if $(G',H',k+1)$ is a YES instance of $\mathcal{D}_d$-CF-ECT.

In the forward direction, let S be a solution to $\mathcal{D}_d$-DCF-ECT in (G,H,R,k). We claim that $S'=S\cup\{v\}$ is a solution to $\mathcal{D}_d$-CF-ECT in $(G',H',k+1)$. Since, G' contains a $(k+2)$-even flower at $v\in V(G')$, v belongs to any solution of size at most $k+1$. Since, $G'-v$ is identical to G, $G'-S'$ does not contain any even cycle. Since, $S\cap R=\emptyset$, $S\cup\{v\}$ is an independent set in H'. This completes the proof in the forward direction.

In the reverse direction, let (G', H', k') be a YES instance of $\mathcal{D}_d$-CF-ECT and S be one of its solution. As argued above, $v \in S$. We claim that $S' = S \setminus \{v\}$ is a solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). Clearly, $G' - v$ is identical to G, therefore, $G - S'$ does not contain any even cycle. Since, $v \in S$, none of the vertices of R belongs to S. Since, H is an induced subgraph of H', S' is an independent set in H. Hence, S' is a solution to $\mathcal{D}_d$-DCF-ECT in (G, H, R, k). ◀

We obtain the following result by Theorem 46 and Lemma 45.

▶ **Theorem 46.** $\mathcal{D}_d$-CF-ECT *admits a kernel with* $\mathcal{O}(d^{\mathcal{O}(d)}k^{\mathcal{O}(d^2)})$ *vertices, when* H *is* d*-degenerate graph.*