Kernels for Structural Parameterizations of Vertex Cover – Case of Small Degree Modulators

Diptapriyo Majumdar, Venkatesh Raman, and Saket Saurabh

The Institute of Mathematical Sciences
Chennai, India
{diptapriyom|vraman|saket}@imsc.res.in

Abstract

Vertex Cover is one of the most well studied problems in the realm of parameterized algorithms and admits a kernel with $\mathcal{O}(\ell^2)$ edges and 2ℓ vertices. Here, ℓ denotes the size of a vertex cover we are seeking for. A natural question is whether Vertex Cover admits a polynomial kernel (or a parameterized algorithm) with respect to a parameter k, that is, provably smaller than the size of the vertex cover. Jansen and Bodlaender [STACS 2011, TOCS 2013] raised this question and gave a kernel for Vertex Cover of size $\mathcal{O}(f^3)$, where f is the size of a feedback vertex set of the input graph. We continue this line of work and study Vertex Cover with respect to a parameter that is always smaller than the solution size and incomparable to the size of the feedback vertex set of the input graph. Our parameter is the number of vertices whose removal results in a graph of maximum degree two. While vertex cover with this parameterization can easily be shown to be fixed-parameter tractable (FPT), we show that it has a polynomial sized kernel.

The input to our problem consists of an undirected graph G, $S \subseteq V(G)$ such that $|S| = k$ and $G[V(G) \setminus S]$ has maximum degree at most 2 and a positive integer ℓ. Given (G, S, ℓ), in polynomial time we output an instance (G', S', ℓ') such that $|V(G')| \leq \mathcal{O}(k^5)$, $|E(G')| \leq \mathcal{O}(k^6)$ and G has a vertex cover of size at most ℓ if and only if G' has a vertex cover of size at most ℓ'. When $G[V(G) \setminus S]$ has maximum degree at most 1, we improve the known kernel bound from $\mathcal{O}(k^3)$ vertices to $\mathcal{O}(k^2)$ vertices (and $\mathcal{O}(k^3)$ edges). In general, if $G[V(G) \setminus S]$ is simply a collection of cliques of size at most d, then we transform the graph in polynomial time to an equivalent hypergraph with $\mathcal{O}(k^d)$ vertices and show that, for $d \geq 3$, a kernel with $\mathcal{O}(k^{d-\epsilon})$ vertices is unlikely to exist for any $\epsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Parameterized Complexity, Kernelization, expansion lemma, vertex cover, structural parameterization

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.331

1 Introduction and Motivation

In the early years of parameterized complexity and algorithms, problems were almost always parameterized by the solution size. Recent research has focussed on other parameterizations based on structural parameters in the input [9], or above or below some guaranteed optimum values [13, 14, 18]. The reasons are many. Such 'non-standard' parameters are more likely to be small in practice. Also, once a problem is shown to be fixed-parameter tractable (FPT) or to have a polynomial sized kernel by a parameterization, it is natural to ask whether the problem is FPT (and admits polynomial kernel) when parameterized by a provably smaller parameter. In the same vein, if we show that a problem is W-hard under a parameterization, it is natural to ask whether it is FPT when parameterized by a provably larger parameter.

10th International Symposium on Parameterized and Exact Computation (IPEC 2015).
Editors: Thore Husfeldt and Iyad Kanj; pp. 331–342
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One of the earliest papers in the realm of alternate parameterization dates back to 1981. Let $\mathfrak{O}_k$ denote the set of all graphs G such that the length of the longest odd cycle is upper bounded by k. Hsu et al. [15] initiated a study of NP-hard optimization problems on $\mathfrak{O}_k$. In particular, they studied the effect of avoiding long odd cycle for the Maximum Independent Set problem and showed that a maximum sized independent set on a graph $G \in \mathfrak{O}_k$ on n vertices can be found in time $n^{\mathcal{O}(k)}$. Later, Grötschel and Nemhauser [12] did a similar study for Max-Cut and obtained an algorithm with running time $n^{O(k)}$ on a graph $G \in \mathfrak{O}_k$ on n vertices. These algorithms, using modern techniques, can be made FPT and also shown to not admit polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [21]. Later Cai [4] did a similar study for Coloring problems. Fellows et al. [10] studied alternate parameterizatons for problems that were proven to be intractable with respect to standard parameterizations. This led to the whole new ecology program and opened up a floodgate of new and exciting research. We refer to [9] for a detailed introduction to the whole program. The kernelization results tend to be harder in this framework, as the rules need to capture the interaction of the structural parameter with the rest of the input, against simply using the property of feasible solutions in the case of the 'standard' parameterization. Vertex Cover, that asks whether a given undirected graph G has a set of size at most ℓ such that $G \setminus S$ is an independent set, for some given integer ℓ, is one of the most well studied problems in the realm of parameterized algorithms and admits an algorithm with running time $1.2738^{\ell} n^{\mathcal{O}(1)}$ and a kernel with $\mathcal{O}(\ell^2)$ edges and 2ℓ vertices [5, 19]. The set S is also called *vertex cover* of the graph. A natural question is whether Vertex Cover admits a polynomial kernel (or a parameterized algorithm) with respect to a parameter k, that is, provably smaller than the size of the vertex cover. Jansen and Bodlaender [16] first raised this question and showed that Vertex Cover admits a kernel of size $\mathcal{O}(f^3)$, where f is the size of the feedback vertex set of the input graph. Since then we have several results in this direction. For Vertex Cover parameterized by the size of the odd cycle transversal and konig vertex deletion set, there is a randomized polynomial sized kernel [17]. Vertex Cover parameterized by the deletion set to chordal graphs or perfect graphs has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [2, 9]. In this paper we continue this line of work on Vertex Cover and study it with respect to a parameter that is always smaller than the solution size and incomparable to the feedback vertex set of the input graph. In particular, we consider the Vertex Cover problem parameterized by the number of vertices whose removal results in a graph of maximum degree at most x, where $x \geq 1$.

Vertex Cover parameterized by degree x Modulator (VC-x-Mod) **Parameter:** k
Input: An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a graph of degree at most x and an integer ℓ.
Question: Does G have a vertex cover of size at most ℓ?

Vertex Cover is known to be NP-complete even on graphs of maximum degree 3 and thus the x in VC-x-Mod must be upper bounded by 2, else we can not even hope to have an algorithm of the form $n^{f(k)}$ for any function f. On the other hand Vertex Cover is polynomial time solvable when the maximum degree is at most 2.

Let G be the input graph along with a vertex subset S such that $|S| \leq k$ and $G[V(G) \setminus S]$ has maximum degree at most 2. We call S a *degree* 2 *modulator of the graph*. By 'guessing' (i.e. trying all possible choices for) the intersection of S with the optimal vertex cover, and solving the remaining problem in polynomial time, we can find a minimum vertex cover of G in $2^k n^{\mathcal{O}(1)}$ time. This shows that VC-2-Mod is FPT. One of our main results is a polynomial kernel for VC-2-Mod.

Our Results. We obtain a kernel for VC-2-Mod with $\mathcal{O}(k^5)$ vertices, and $\mathcal{O}(k^6)$ edges. Our result is in contrast to the fact that Vertex Cover parameterized by treewidth 2 modulator (i.e. when $G[V(G) \setminus S]$ is a general graph of treewidth at most 2) has no polynomial sized kernel unless NP $\subseteq$ coNP/poly [6]. We also address the kernelization question for VC-1-Mod. Here, a kernel with $\mathcal{O}(k^3)$ vertices was already known from the result of [16] for Vertex Cover parameterized by the feedback vertex set size. This follows as the size of the feedback vertex set is at most the size of a degree 1 modulator. We improve the kernel size to $\mathcal{O}(k^2)$ vertices. More generally, we consider the Vertex Cover problem when parameterized by the size of a subset of vertices whose removal results in a graph with all components being cliques of size at most a constant d. We call a graph G *d-cluster graph* if every connected component of G is a clique and has size at most d. In particular we study the following problem:

Vertex Cover parameterized by d-CVD (VC-param-d-CVD) **Parameter:** k
Input: An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a d-cluster graph and an integer ℓ.
Question: Does G have a vertex cover of size at most ℓ?

Observe that VC-1-Mod and VC-param-2-CVD are the same problems. It is known that if the resulting graph is simply a clique (with no bound on the size), then a polynomial sized kernel is unlikely [2]. We show that the input graph of VC-param-d-CVD can be transformed in polynomial time to obtain an equivalent *hypergraph* with $\mathcal{O}(dk^d)$ vertices where each hyperedge is of size at most d. We also show that a kernel with $\mathcal{O}(k^{d-\epsilon})$ vertices, for any $\epsilon > 0$, is unlikely unless NP $\subseteq$ coNP/poly. We think that this idea of using hyperedges to capture certain constraints could find applications while doing a compression for the parameterized problem.

Observe that we have always assumed that the *modulator* is given as a part of the input. However, this constraint can be relaxed as both VC-2-MOD and VC-param-d-CVD admit constant factor approximation algorithms. For example, for VC-2-MOD, there is a factor 4-approximation algorithm and for VC-param-d-CVD, we can get an approximation algorithm with factor $(d+1)$(see [20] for approximation algorithms). However, we can obtain constant factor approximation algorithms can be obtained by greedily finding an obstruction (like a vertex v and any of its three neighbors in the case of VC-2-MOD) and selecting all the vertices in this obstruction to the approximate solution we are constructing. This implies that rather than demanding that modulators are given as a part of the input, we can first compute it using the polynomial time constant factor approximation algorithms and then run our kernelization algorithms using these. These will result in kernels with same asymptotic upper bounds as mentioned above.

2 Preliminaries and Definitions

By $[r]$, we mean the set $\{1, 2, \ldots r\}$. Throughout the paper we denote the *vertex cover number* (the size of a minimum vertex cover) by $vc(G)$.

▶ **Definition 1** (Kernelization). Let $L \subseteq \sum^* \times \mathbb{N}$ be a parameterized language. Kernelization is a procedure that replaces the input instance (I, k) by a reduced instance (I', k') such that $k' \leq k$, $|I'| \leq g(k)$ for some function g depending only on k and $(I, k) \in L$ if and only if $(I', k') \in L$. The reduction from (I, k) to (I', k') must be computable in $poly(|I| + k)$ time.

▶ **Definition 2** (Soundness/Safeness of Reduction Rule). A reduction rule that replaces an instance (I, k) of a parameterized language L by a reduced instance (I', k') is said to be sound or safe if $(I, k) \in L$ if and only if $(I', k') \in L$.

▶ **Definition 3** (Polynomial parameter transformation (PPT)). Let P_1 and P_2 are two parameterized languages. We say that P_1 is polynomial parameter reducible to P_2 if there exists a polynomial time computable function (or algorithm) $f : \sum^* \times \mathbb{N} \to \sum^* \times \mathbb{N}$, a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that $(x, k) \in P_1$ if and only if $f(x, k) \in P_2$ and $k' \leq p(k)$ where $f((x, k)) = (x', k')$. We call f to be a polynomial parameter transformation from P_1 to P_2.

The following proposition gives the use of the polynomial parameter transformation for obtaining kernels for one problem from another.

▶ **Proposition 4** ([3]). *Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems and assume there exists a PPT from P to Q. Furthermore, assume that the classical version of P is NP-hard and Q is in NP. Then if Q has a polynomial kernel implies that P has a polynomial kernel.*

The following powerful variation of Hall's matching theorem, known as Expansion Lemma, is used in some of our reduction rules.

▶ **Lemma 5** (q-Expansion Lemma). [11, 22, 24]) *Let q be a positive integer and G be a bipartite graph with vertex partition A and B such that $|B| > q|A|$ and there are no isolated vertices in B. Then, there exists non-empty subsets $X \subseteq A, Y \subseteq B$ obtainable in polynomial time, such that*

- *there is a q-expansion of X into Y. I.e. there is a $M \subseteq E$ such that every vertex in X is incident with exactly q edges of M. Moreover M saturates exactly $q|X|$ vertices in Y, and*
- *$N_G(Y) \subseteq X$.*

In Section 4, after applying some reduction rules, the input graph gets converted to a hypergraph where hyperedges consisting of more than 2 vertices are sometimes present in $G[S]$. We define an independent set and a vertex cover in hypergraph as follows. Recall that by $G[S]$ for a subset of vertices S (where G is a graph or a hypergraph), we denote the subgraph that consists of the vertices of S and all the (hyper) edges which are completely contained in S.

▶ **Definition 6** (Independent Set in a hypergraph). $A \subseteq V(G)$ is said to be an independent set in a hypergraph if no hyperedge is contained in $G[A]$.

▶ **Definition 7** (Vertex Cover in a hypergraph). $A \subseteq V(G)$ is said to be a vertex cover in hypergraph if for every hyperedge $e \in E(G)$, $A \cap V(e) \neq \emptyset$ where $V(e)$ be the set of vertices present in the hyperedge e.

A vertex cover in a hypergraph is also known as a *hitting set*.

3 Kernel for VC-2-Mod

Throughout this section for an input (G, S, ℓ) to VC-2-Mod we use F to denote $V(G) \setminus S$. Now, we are ready to describe the reduction rules that compress $G[F]$ to an equivalent instance whose size is polynomial in k. Note that rules will have to be applied sequentially, and after every rule is applied, we need to start from the beginning and exhaustively apply applicable rules; some of the earlier rules may become applicable after a rule is applied. We allow the input graphs to have self loops. The main reason for this is that even though input graph may not have self loops, some reduction rules (for example, Reduction Rule 8) may create self loops.

3.1 Ensuring minimum degree 3

The following reduction rules are standard for the Vertex Cover problem (see, for example, Chapter 4 of [8] for correctness of the rules).

▶ **Reduction Rule 1.** *Remove isolated vertices from* G*.*

▶ **Reduction Rule 2.** *If* $\exists u \in V(G)$ *such that there is a self loop with* u*, then* $G' \leftarrow G \setminus \{u\}, \ell' \leftarrow \ell - 1$*.*

▶ **Reduction Rule 3.** *If* $\exists u \in G$ *such that* $deg_G(u) = 1$ *and* v *is its unique neighbour, then* $G' \leftarrow G \setminus \{u, v\}, \ell' \leftarrow \ell - 1$*.*

▶ **Reduction Rule 4.** *If* $\exists u \in G$ *such that* $deg_G(u) = 2$ *and let* v, w *be its* 2 *neighbours in* F*, then do the followings:*

- *If* $(v, w) \in E(G)$*, then* $G' \leftarrow G \setminus \{u, v, w\}, \ell' \leftarrow \ell - 2$
- *If* $(v, w) \notin E(G)$*, then* $G' \leftarrow G \setminus \{u, v, w\} \cup \{u_{new}\}, \ell' \leftarrow \ell - 1$*, and make all vertices adjacent to* v *and* w *(except* u*) in* G *adjacent to* u_{new}*.*

When the above reduction rules are not applicable, the minimum degree in the graph is at least 3, and hence every vertex $v \in F$ has at least one neighbour in S. We partition F into F_0, F_1 and F_2 such that every connected component of $G[F_0]$ is an isolated vertex, every connected component of $G[F_1]$ is either a path (of length at least 2) or a cycle of even length (length at least 4) and every connected component of $G[F_2]$ is an odd cycle. As every component in $G[F_2]$ is an odd cycle, we interchangeably use the term component or an odd cycle to mean the same thing in $G[F_2]$. Central to the rules in this subsection is a notion of a *blocking set*, we define the notion first and then prove some properties about them.

3.2 Blocking Sets and their Properties

▶ **Definition 8** (Blocking Set and Good Set)**.** Let $B \subseteq V(G)$. We call B to be a *blocking set* if $vc(G[V(G) \setminus B]) + |B| > vc(G)$. We call a *blocking set* B to be a *minimal blocking set* if no proper subset of B is a blocking set. A set $B \subseteq F$ is called a *good set* if it is not a *blocking set*.

If an algorithm picks the vertices of a blocking set B into a solution, then any way to complete it to an optimum vertex cover results in a non-optimal solution. So the blocking set 'blocks' the completion step from resulting into an optimal solution. For example, in a cycle of even length, the end points of any edge form a blocking set as no optimum solution for the cycle contains two vertices of the same edge. We will apply these notions to $G[F]$.

For two vertices a_i, a_j of a cycle C where vertices are ordered as $a_0, a_1, \ldots, a_{|C|-1}$ with $i < j \mod |C|$, by $dist(a_i, a_j)$ we mean the length (the number of edges) in the *clockwise* path that goes through the vertices $a_{i+1}, a_{i+2}, \ldots, a_{j-1}$ where all the subscripts are taken mod $|C|$. The following statement is easy to verify.

▶ **Observation 9.** *In a cycle, no single vertex forms a blocking set; hence minimal blocking sets are of size at least* 2 *in cycles.*

▶ **Lemma 10** ($\star$[1])**.** *Let* $B \subseteq V(F)$ *be a minimal blocking set in* $G[F]$*. Then there exists a unique* C *for which* $B \subseteq V(C)$ *where* C *is a component of* $G[F]$*.*

[1] Due to lack of space, the proofs of results marked $\star$, and the proof of correctness and the polynomial runtime of our reduction rules will appear in the full version.

▶ **Theorem 11** (⋆).
- *In an odd cycle, the only minimal blocking sets are of size* 3 *where the clockwise distance between every pair of them is odd.*
- *In an even cycle, the only minimal blocking sets are of size* 2 *where the clockwise distance between every pair of them is odd.*

▶ **Definition 12** (Bad Component and Nice Component)**.** Let C be a cycle in $G[F_2]$. If there exists an independent set $A \subseteq S$ of size at most 3 such that $N_G(A) \cap C$ contains a blocking set, then we call C a *bad component.* A component is said to be a *nice component* if it is not a bad component.

We partition the set of bad components (in F_2) as $\mathcal{B}_1, \mathcal{B}_2$ and $\mathcal{B}_3$ where
$\mathcal{B}_1 = \{C|C$ is a component and $\exists x \in S$ such that $N_G(x) \cap C$ contains a blocking set$\}$.
$\mathcal{B}_2 = \{C|C$ is a component and $\exists x, y \in S, (x, y) \notin E(G)$ such that $C \cap (N_G(x) \cup N_G(y))$ contains a blocking set$\} \setminus \mathcal{B}_1$.
$\mathcal{B}_3 = \{C|C$ is a component and $\exists x, y, z \in S, \{x, y, z\}$ is independent set such that $C \cap (N_G(x) \cup N_G(y) \cup N_G(z))$ contains a blocking set$\} \setminus (\mathcal{B}_1 \cup \mathcal{B}_2)$.

By $\mathcal{B}_4$ we denote the set of nice components. The following observation follows from the definitions.

▶ **Observation 13.** *If $C \in \mathcal{B}_2$ then for any blocking set B in C, $B \not\subset N_G(x)$ for any $x \in S$ and if $C \in \mathcal{B}_3$, then for any blocking set B in C, $B \not\subset N_G(A)$ for any independent set of size at most* 2 *in S.*

3.3 Towards bounding the number of components

In this subsection we describe three rules, two of which are powerful to help us bound the number of components in $G[F]$.

▶ **Reduction Rule 5** (NiceComponent Rule)**.** *Let C be a nice component in F. Then $G' \leftarrow G \setminus C, \ell' \leftarrow \ell - vc(G[C])$.*

▶ **Reduction Rule 6.** *If there exists a vertex $x \in S$ such that $vc(G[F \setminus N_G(x)]) + |N_G(x) \cap F| \geq vc(G[F]) + |S| + 1$, then $G' \leftarrow G \setminus \{x\}, \ell' \leftarrow \ell - 1$.*

▶ **Reduction Rule 7.** *If there exists $x, y \in S, (x, y) \notin E(G)$ such that $vc(G[F \setminus N_G(\{x, y\})]) + |N_G(\{x, y\}) \cap F| \geq vc(G[F]) + |S| + 1$, then add edge (x, y) into G.*

Note that while this rule does not decrease the size of the graph or ℓ, it does enable the applicability of some rules (for example Reduction Rule 8 which is stated later).

▶ **Lemma 14.** *If Reduction Rules 6 and 7 are not applicable, then the following statements are true. Let M be a maximum matching of $G[F_1]$, and let M_1 be a maximum matching of $G[F_2]$.*

1. *Then, for every $x \in S$,*
   - $|N_G(x) \cap F_0| \leq |S|$.
   - *$N_G(x) \cap F$ contains both end points of at most $|S|$ edges of M.*
   - *$N_G(x) \cap F$ contains both end points of at most $|S| + c$ edges of M_1 where c is the number of odd cycles in $G[F_2]$.*
   - *$N_G(x) \cap F$ contains a blocking set in at most $|S|$ cycles in $\mathcal{B}_1$.*
2. *For every pair x, y of vertices in S such that $(x, y) \notin E(G)$,*
   - *$N_G(\{x, y\}) \cap F$ contains both end points of at most $|S|$ edges of M.*

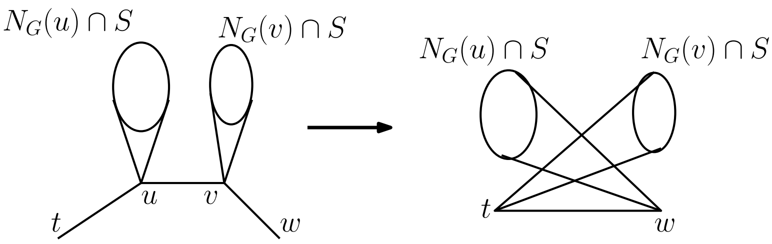

Figure 1 An Illustration of Reduction Rule 8.

- *$N_G(\{x, y\}) \cap F$ contains both end points of at most $|S| + c$ edges of M_1 where c is the number of odd cycles in $G[F_2]$.*
- *$N_G(\{x, y\}) \cap F$ contains blocking set in at most $|S|$ cycles in $\mathcal{B}_2$.*

Proof sketch. The set F_0 contains isolated vertices. Therefore, if a vertex $x \in S$ is adjacent to at least $|S| + 1$ vertices in F_0, then any vertex cover C such that $x \notin C$ must pick $vc(F)$ vertices from F (which do not contain any vertex from F_0) and at least $|S| + 1$ vertices from F_0. Therefore, Reduction Rule 6 becomes applicable. Hence, $|N_G(x) \cap F_0| \leq |S|$. By similar arguments and using properties of blocking sets in odd cycles, we can prove the other facts when Reduction Rule 6 is not applicable.

The graph $G[F_1]$ is bipartite and let $vc(G[F_1]) = |M|$. Now, any vertex cover C such that $x, y \notin C$ with $(x, y) \notin E(G), x, y \in S$ must contain $vc(F)$ vertices from F containing exactly one endpoint from every edge of M and at least $|S| + 1$ other vertices from those edges of M both of whose end points are present in $N_G(\{x, y\})$. Therefore, when reduction rule 7 is not applicable, then $N_G(\{x, y\})$ must contain both end points of at most $|S|$ edges in M. By using similar arguments and propertes of blocking sets in odd cycles, we can prove the other facts when reduction rule 7 is not applicable. ◀

▶ **Corollary 15.** *When Reduction Rules 6, 7 are not applicable, then*
- $|\mathcal{B}_2| \leq k\binom{k}{2}$.
- $|\mathcal{B}_1| \leq k^2$.
- $|F_0| \leq k^2$.

Proof sketch. We know that for every odd cycle $C \in \mathcal{B}_1$, there exists a vertex $x \in S$ such that $N_G(x)$ contains a blocking set in C. By Lemma 14, for every $x \in S$, there are at most k cycles in $\mathcal{B}_1$ such that $N_G(x)$ contains a blocking set in each of those cycles. Therefore, $|\mathcal{B}_1| \leq k^2$. By using a similar argument, we can justify the other claims.

Another easy consequence of Lemma 14 is that $|F_0| \leq k^2$. ◀

By a similar argument, we can bound $|\mathcal{B}_3|$ as well, but we take this up in Section 3.5 by a different argument which gives a better bound.

3.4 Bounding the number of vertices in F_1

Now we describe a rule that helps bound the number of vertices in F_1. Here instead of bounding the number of components in F_1 and the number of vertices in each component, we directly bound the number of edges in the maximum matching M of $G[F_1]$. The rule is a slight variation of one proposed by Jansen and Bodlaender [16] for Vertex Cover parameterized by the feedback vertex set number.

▶ **Reduction Rule 8** (Edge Rule). *Let* $\exists(u,v) \in E(G[F])$ *such that* $(N_G(u) \cap S) \cap (N_G(v) \cap S) = \emptyset$ *and* $\forall x \in N_G(u) \cap S, \forall y \in N_G(v) \cap S : (x,y) \in E(G)$ *(i.e.* $N_G(u) \cap S$ *and* $N_G(v) \cap S$ *induce a complete bipartite graph). (See Figure 1 for an illustration.) Then do the following.*

- *Delete* u, v *from the graph.*
- *If* u *has a neighbour* t *in* F *which is not* v*, then make* t *adjacent to every vertex in* $N_G(v) \cap S$*.*
- *If* v *has a neighbour* w *in* F *which is not* u*, then make* w *adjacent to every vertex in* $N_G(u) \cap S$*.*
- *If the vertices* t, w *exist, then they are unique and add the edge* (t, w)*.*
- *Set* ℓ' *to* $\ell - 1$*.*

Then we have the following lemma.

▶ **Lemma 16** (⋆). *When reduction rules 1 to 8 are not applicable* $G[F_1 \cup F_0]$ *has* $\mathcal{O}(k^3)$ *vertices.*

3.5 Bounding the the number of odd cycles

We use the Expansion Lemma 5 to get an upper bound on $\mathcal{B}_3$. We construct a bipartite graph as $H = (S_3, \mathcal{B}_3, E)$ where S_3 is the set of all independent sets of size 3 from S, and $E(H)$ is defined as follows. $E(H) = \{(I, L) | \exists B \subseteq V(L)$ such that B is a blocking set of size 3 and $B \subseteq N_G(I)\}$.

▶ **Reduction Rule 9.** *If* $|\mathcal{B}_3| > 5|S_3|$*, then apply Expansion lemma 5 with* $q = 5$ *from* S_3 *to* $\mathcal{B}_3$ *to get* $A \subseteq S_3, B \subseteq \mathcal{B}_3$ *such that* $N_H(B) \subseteq A$ *and there is a* 5*-expansion from* A *to* B*.*

Associated with every $(x,y,z) \in A$*, there are* 5 *distinct cycles in* B*. Pick one of those* 5 *cycles for each such* $\{x,y,z\} \in A$*. Let* $C_{p_1}, \ldots, C_{p_{|A|}}$ *be collection of such cycles. Then set* $G' \leftarrow G \setminus (C_{p_1} \cup \ldots \cup C_{p_{|A|}}), \ell' \leftarrow \ell - (\sum_{i=1}^{|A|} vc(C_{p_i}))$*.*

Now it is easy to show that

▶ **Corollary 17.** *If Reduction Rule 9 is not applicable, then* $|\mathcal{B}_3| \leq 5\binom{k}{3}$*.*

Finally the following lemma follows from Reduction Rule 5 (as if this rule is not applicable, every component is bad) and Corollaries 15 and 17.

▶ **Lemma 18.** *If none of the above reduction nules is applicable, then the number of components in* $G[F_2]$ *is* $\mathcal{O}(k^3)$*.*

3.6 Bounding $G[F_2]$ and Putting things together

▶ **Lemma 19.** *When Reduction Rules 1 to 9 are not applicable, the number of vertices in* $G[F_2]$ *is* $\mathcal{O}(k^5)$*.*

Proof. By Reduction Rule 8, for every edge $(u,v) \in E(G[F])$, there is either a vertex $x \in S$ such that $x \in N(u) \cap N(v)$ or there exists a pair of non-adjacent vertices $x, y \in S$ such that $x \in N(u), y \in N(v)$. In the former case, we associate the vertex x with the edge (u,v) and to the pair of vertices (x,y) in the case of the latter. By Lemma 14 every $x \in S$ is adjacent to both end points of at most $k+c$ edges of M_1 in C, and every pair of non-adjacent vertices $x, y \in S$, are together adjacent to both end points of at most $k+c$ edges of M_1. Therefore, $|M_1| \leq (k + \binom{k}{2})(k+c) = \mathcal{O}(k^5)$ as c is $\mathcal{O}(k^3)$. It follows that $|V(G[F_2])| = 2|M_1| + c = \mathcal{O}(k^5)$. ◀

The following theorem is an immediate consequence of Lemma 19 and Lemma 16. The bound on the number of edges follows because the number of edges in $G[S]$ is $\mathcal{O}(k^2)$, the number of edges in $G[F]$ is $\mathcal{O}(k^5)$ (as each vertex has degree at most 2 in $G[F]$, and the number of edges between F and S can be at most $\mathcal{O}(k^6)$.

▶ **Theorem 20.** VC-2-Mod *has a kernel consisting of* $\mathcal{O}(k^5)$ *vertices and* $\mathcal{O}(k^6)$ *edges.*

4 Vertex Cover parameterized by bounded cluster vertex deletion set

Now we consider the Vertex Cover problem when parameterized by the size of the degree 1 modulator. Here the resulting graph after removal of the modulator is a collection of isolated vertices and edges – i.e. a collection of cliques of size at most 2. In fact, we will consider the general problem VC-param-d-CVD.

If there is no bound on the sizes of the cliques in $G[F]$, then it is known that the problem has no polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This follows from a result of Bodlaender et al. [2] who showed this infeasibility of polynomial kernel for Vertex Cover when parameterized by *clique deletion set* which is a set of vertices whose removal results in a clique.

We start with a definition, similar to the notion of *Bad and Nice component* in the kernelization of VC-2-Mod.

▶ **Definition 21** (Bad Clique and Nice Clique)**.** A clique C of $G[F]$ is said to be a *bad clique* if $\exists A \subseteq S$ such that A is independent and $|A| \leq d$ and $V(C) \subseteq N_G(A)$. A clique is said to be a *nice clique* if it is not a *bad clique*.

Observe that any clique C in $G[F]$, that contains a vertex which has no neighbour in S, is a nice clique. Now we proceed to state the list of reduction rules for this problem. As a preprocessing, we only require that isolated vertices are removed (i.e. there is no need to even make the graph minimum degree 3 using the rules of Section 3.1).

▶ **Reduction Rule 10.** *For every nice clique C of $G[F]$, delete it to obtain $G' \leftarrow G \setminus V(C)$ and make $\ell' \leftarrow \ell - (|V(C)| - 1)$.*

Note that when Reduction rule 10 is not applicable, every vertex in F has a neighbour in S.

▶ **Reduction Rule 11.** *Let $\exists I \subseteq S$ such that $|I| \leq d-1, I$ is an independent set and $N_G(I)$ contains all vertices of at least $|S|+1$ cliques in F, then do the following:*

- *If $|I| = 1$, then $G' \leftarrow G \setminus I, \ell' \leftarrow \ell - 1$.*
- *If $2 \leq |I| \leq (d-1)$, then add the hyperedge $\{x_1, \ldots, x_{d-1}\}$ into G where $I = \{x_1, \ldots, x_{d-1}\}$.*

We partition the set of bad cliques in $G[F]$ into 2 parts as follows.
$Z_1 = \{Z | Z \text{ such that } \exists I \subseteq S, |I| \leq d-1, V(Z) \subseteq N_G(I)\}$.
$Z_2 = \{Z | Z \text{ such that } \exists I \subseteq S, |I| = d, V(Z) \subseteq N_G(I)\} \setminus Z_1$.
Note that hyperedges consisting of more than 2 vertices are present only in S. Therefore, for any vertex $x \in S$, $N_G(x) \cap F = \{y \in F | (x,y) \in E(G)\}$ and for any vertex $u \in F$, $N_G(u) \cap S = \{v \in S | (u,v) \in E(G)\}$. For every vertex x, let $HE(x) = \{e \in E(G) | x \in V(e)\}$. For every hyperedge e let $V(e)$ be the set of vertices that are present in the hyperedge e. The following two rules are due to [1].

▶ **Reduction Rule 12** (Vertex and Edge Domination Rule)**.** *Let there be two vertices x, y such that $HE(x) \subseteq HE(y)$, then delete x from G. Similarly, if there are two hyperedges e_1, e_2 such that $V(e_1) \subseteq V(e_2)$, then delete e_2 from G.*

▶ **Corollary 22.** *When none of the above reduction rules are applicable,* $|Z_1| \leq k(\sum_{i=1}^{d-1} \binom{k}{i})$.

Proof. By the definition of Z_1, for every clique $C \in Z_1$, there exists an independent set X of size at most $d-1$ of S such that $N_G(X) \subseteq C$. By Reduction Rule 11, for every independent set X of size at most $d-1$, $N_G(X)$ contains all end points of at most k cliques. Therfore, the number of cliques in Z_1 is at most $k(\sum_{i=1}^{d-1} \binom{k}{i})$. ◀

Using similar arguments, we can give a bound for $|Z_2|$ of $\mathcal{O}(k^{d+1})$, but we give an improved bound using the expansion lemma. Let Z be a clique in Z_2 consisting of d vertices. For every $u, v \in V(Z)$, we have that $N_G(u) \cap N_G(v) \cap S = \emptyset$ by definition. But for every $u \in V(Z)$, we have $N_G(u) \cap S \neq \emptyset$. Now, we construct a bipartite graph $H(S_B, Z_2, J)$. Let $S_B = \{X \subseteq S | X$ is an independent set in G and $|X| = d\}$. We add an edge (I, Z) in J if $V(Z) \subseteq N_G(I)$.

▶ **Reduction Rule 13.** *If* $|Z_2| > (d+1)|S_B|$, *then apply Expansion Lemma 5 with* $q = (d+1)$ *from* S_B *to* Z_2 *to obtain* $P_B \subseteq S_B, Q_B \subseteq Z_2$ *such that* $N_H(Q_B) \subseteq P_B$. *For every* $X \in S_B$, *add the hyperedges* $\{x_1, \ldots, x_d\}$ *where* $X = \{x_1, \ldots, x_d\}$.

▶ **Theorem 23.** VC-PARAM-d-CVD *has a compression of size* $\mathcal{O}(k^d)$. *In other words, when no reduction rule is applicable, the resulting hypergraph has* $\mathcal{O}(k^d)$ *vertices. Each hyperedge is of size at most* d, *and hyperedges of size more than* 2 *are present only in* S.

Proof. By Reduction Rule 10, every clique $Z \in G[F]$ is a bad clique. Every bad clique is of 2 types. By Corollary 22, $|Z_1| \leq k(\sum_{i=1}^{d-1} \binom{k}{i})$. Now for every clique Z of Z_2, there exists an independent set $I \subseteq S$ of size d such that $N_G(I) \subseteq Z$. There are at most $\binom{k}{d}$ independent sets of size d in S. As Reduction Rule 13 is not applicable, $|Z_2| \leq (d+1)\binom{k}{d}$. Therefore, the resulting hypergraph has a kernel of size at most $k(\sum_{i=1}^{d-1} \binom{k}{i}) + (d+1)\binom{k}{d} = \mathcal{O}(k^d)$. ◀

When $G[F]$ is a graph of degree at most 1, every component of $G[F]$ is either an isolated vertex or an edge. Setting $d = 2$ in the above theorem, we get the following. The edge bound follows as every vertex in F has degree at most 1 within F and at most k into S. Note also that the resulting hypergraph is simply a graph.

▶ **Corollary 24.** VC-1-MOD *has a kernel on* $\mathcal{O}(k^2)$ *vertices and* $\mathcal{O}(k^3)$ *edges.*

5 Lower Bounds

Now, we prove a lower bound, under complexity theoretic assumptions for the size of the kernel of the problems we considered in this paper. We prove this by giving a polynomial parameter transformation (see definition in Section 2) from d-CNF-SAT to our problem(s) and use the following theorem due to Dell and Melkebeek [7]. Here d-CNF-SAT is the problem of testing the satisfiability of a d-CNF formula, a boolean formula where the clauses are in CNF form with at most d variables each.

▶ **Theorem 25** (Lower Bound for d-CNF-SAT)**.** d-CNF-SAT *parameterized by* n, *the number of variables, has no kernel of size* $\mathcal{O}(n^{d-\epsilon})$ *for any* $d \geq 3, \epsilon > 0$ *unless* NP $\subseteq$ coNP/poly.

There is a standard reduction (see for example [23], when $d = 3$) from d-CNF-SAT to VC-PARAM-d-CVD.

▶ **Theorem 26** (⋆). *There exists a polynomial parameter transformation from the* d-CNF-SAT *parameterized by the number of variables to* VC-PARAM-d-CVD. *In* VC-PARAM-d-CVD, *the size of the modulator is twice the number of variables in the* d-CNF-SAT *formula.*

The following theorem follows from Theorem 26 and Proposition 4.

▶ **Theorem 27.** VC-PARAM-d-CVD *has no kernel of size* $\mathcal{O}(k^{d-\epsilon})$ *for any* $d \geq 3, \epsilon > 0$ *unless* NP ⊆ coNP/poly.

As a collection of cliques of size at most 3 is a subclass of graphs with degree at most 2, we have the following corollary.

▶ **Corollary 28.** VC-2-MOD *has no kernel of size* $\mathcal{O}(k^{3-\epsilon})$ *unless* NP ⊆ coNP/poly.

6 Conclusion

In this paper we gave a polynomial kernel for VC-2-MOD. There is a gap between upper and lower bounds on the kernel sizes we obtained; it would be interesting to bridge this gap. It is known that VERTEX COVER admits a randomized polynomial kernel parameterized by the odd cycle transversal number of the graph (minimum number of vertices whose deletion results in a bipartite graph). Is it possible to obtain a deterministic kernel for this parameterization (maybe using some ideas from this paper)? We think this might be easier and probably the first step towards obtaining a deterministic kernel for the ODD CYCLE TRANSVERSAL problem itself.

Acknowledgements. We thank the referees whose comments helped unify some of the reduction rules and improved the presentation of the paper.

References

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

2 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.

3 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

4 Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

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

6 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On the hardness of losing width. *Theory Comput. Syst.*, 54(1):73–82, 2014.

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

8 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

9 Michael R. Fellows, Bart M. P. Jansen, and Frances A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *Eur. J. Comb.*, 34(3):541–566, 2013.

10 Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Matthias Mnich, Frances A. Rosamond, and Saket Saurabh. The complexity ecology of parameters: An illustration using bounded max leaf number. *Theory Comput. Syst.*, 45(4):822–848, 2009.

11 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and kernelization. In *STACS*, pages 189–200, 2011.

12 Martin Grötschel and George L. Nemhauser. A polynomial algorithm for the max-cut problem on graphs without long odd cycles. *Math. Programming*, 29(1):28–40, 1984.

13 Gregory Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. A probabilistic approach to problems parameterized above or below tight bounds. *J. Comput. Syst. Sci.*, 77(2):422–429, 2011.

14 Gregory Gutin and Anders Yeo. Constraint satisfaction problems parameterized above or below tight bounds: A survey. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 257–286. Springer, 2012.

15 Wen Lian Hsu, Yoshiro Ikura, and George L. Nemhauser. A polynomial algorithm for maximum weighted vertex packings on graphs without long odd cycles. *Math. Programming*, 20(2):225–232, 1981.

16 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013.

17 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459, 2012.

18 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms*, 11(2):15:1–15:31, 2014.

19 George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8(1):232–248, 1975.

20 Michael Okun and Amnon Barak. A new approach for approximating node deletion problems. *Inf. Process. Lett.*, 88(5):231–236, 2003.

21 Fahad Panolan and Ashutosh Rai. On the kernelization complexity of problems on graphs without long odd cycles. In *COCOON 2012*, volume 7434 of *LNCS*, pages 445–457. Springer, 2012.

22 Elena Prieto. *Systematic kernelization in FPT algorithm design*. PhD thesis, The University of Newcastle, Australia, 2005.

23 Michael Sipser. *Introduction to the theory of computation.* PWS Publishing Company, 1997.

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