



Structural Parameterizations of Dominating Set Variants

Dishant Goyal¹, Ashwin Jacob^{2(✉)}, Kaushtubh Kumar³,
Diptapriyo Majumdar², and Venkatesh Raman²

¹ Indian Institute of Technology Delhi, New Delhi, India
dishant.in@gmail.com

² The Institute of Mathematical Sciences, HBNI, Chennai, India
{ajacob, diptapriyom, vraman}@imsc.res.in

³ Mentor Graphics, Noida, India
kaustubhkp10@gmail.com

Abstract. We consider structural parameterizations of the fundamental dominating set problem and its variants in the parameter ecology program. We give improved fixed-parameter tractable (FPT) algorithms and lower bounds under well-known conjectures for dominating set in graphs that are k vertices away from a cluster graph or a split graph. These are graphs in which there is a set of k vertices (called the modulator) whose deletion results in a cluster graph or a split graph. We also call k as the deletion distance (to the appropriate class of graphs). Specifically, we show the following results. When parameterized by the deletion distance k to cluster graphs,

- we can find a minimum dominating set in $\mathcal{O}^*(3^k)$ time (\mathcal{O}^* notation ignores polynomial factors of input). Within the same time, we can also find a minimum independent dominating set (IDS) or a minimum efficient dominating set (EDS) or a minimum total dominating set. These algorithms are obtained through a dynamic programming approach for an interesting generalization of set cover which may be of independent interest.
- We complement our upper bound results by showing that at least for dominating set and total dominating set, $\mathcal{O}^*((2-\epsilon)^k)$ time algorithm is not possible for any $\epsilon > 0$ under, what is known as, Set Cover Conjecture. We also show that most of these variants of dominating set do not have polynomial sized kernel.

The standard dominating set and most of its variants are NP-hard or W[2]-hard in split graphs. For the two variants IDS and EDS that are polynomial time solvable in split graphs, we show that when parameterized by the deletion distance k to split graphs,

- IDS can be solved in $\mathcal{O}^*(2^k)$ time and we provide an $\Omega(2^k)$ lower bound under the strong exponential time hypothesis (SETH);
- the 2^k barrier can be broken for EDS by designing an $\mathcal{O}^*(3^{k/2})$ algorithm. This is one of the very few problems with a runtime better than $\mathcal{O}^*(2^k)$ in the realm of structural parameterization. We also show that no $2^{o(k)}$ algorithm is possible unless the exponential time hypothesis (ETH) is false.

1 Introduction

1.1 Motivation

The DOMINATING SET problem is one of the classical NP-Complete graph theoretic problems. It asks for a minimum set of vertices in a graph such that every vertex is either in that set or has a neighbor in that set. It, along with several variations including *independent domination*, *total domination*, *efficient domination*, *connected domination*, *total perfect domination*, *threshold domination* are well-studied in all algorithmic paradigms including parameterized complexity and approximation and in structural points of view. All of these versions are hard for the parameterized complexity class $W[1]$ in general graphs when parameterized by solution size [12] and hence is unlikely to be fixed-parameter tractable (See [8] for more details).

One of the goals in parameterized complexity is to identify parameters under which (even hard) problems are fixed-parameter tractable. This is also of practical interest as often there are some small parameters (other than solution size) that capture important practical inputs. This has resulted in the parameter ecology program where one studies problems under a plethora of parameters and recently there has been a lot of active research [4, 13, 18] in this area. In particular, identifying a parameter as small as possible, under which a problem is fixed-parameter tractable or has a polynomial sized kernel is an interesting direction of research. We continue this line of research and consider parameterizations of DOMINATING SET variants that are more natural and functions of the input graph. *Structural parameterization* of a problem is where the parameter is a function of the input structure rather than the standard output size. To the best of our knowledge, this is the first serious study of structural parameterization of any version of the dominating set problem.

Our parameter of interest is the ‘distance’ of the graph from a natural class of graphs. Here by distance we mean the number of vertices whose deletion results in the class of graphs. Note that if dominating set is NP-hard in a graph class, then it will continue to be NP-hard even on graphs that are k away from the class, even for constant k (in particular for $k = 0$) and hence is unlikely to be fixed-parameter tractable. Hence it is natural to consider graphs that are not far from a class of graphs where DOMINATING SET is polynomial time solvable. Our case study considers two such special graphs: cluster graphs where each connected component is a clique and split graphs where the vertex set can be partitioned into a clique and an independent set. In the former, all the variants of dominating set we consider are polynomial time solvable, while in the latter class of split graphs, we consider independent and efficient dominating set that are polynomial time solvable. We call the set of vertices whose deletion results in a cluster graph and split graph as *cluster vertex deletion set* (CVD) and *split vertex deletion set* (SVD) respectively.

Finally, we remark that the size of minimum CVD and minimum SVD are at most the size of a minimum vertex cover in a graph, which is a well-studied parameterization in the parameter-ecology program [13].

1.2 Definitions, Our Results and Organization of the Paper

We start with describing the variants of dominating set we consider in the paper. A subset $S \subseteq V(G)$ is a *dominating set* if $N[S] = V(G)$. If S is an independent set, then S is an *independent dominating set*. It is called an *efficient dominating set* if for every vertex $v \in V$, $|N[v] \cap S| = 1$. Note that an efficient dominating set may not exist for a graph (for example, for a 4-cycle). If for every vertex v , $|N(v) \cap S| \geq r$, S is a *threshold dominating set* with threshold r . When $r = 1$, S is a *total dominating set*. Note that for dominating set, the vertices in S do not need other vertices to dominate them, but they do in a total dominating set. For more on these dominating set variants, see [15]. We will often denote *dominating set*, *efficient dominating set*, *independent dominating set*, *total dominating set* and *threshold dominating set* by DS, EDS, IDS, TDS and ThDS respectively in the rest of the article. When we say that a graph G is k -away from a graph in a graph class, what we mean is that there is a subset S of k vertices in the graph such that $G \setminus S$ belongs to the class.

Now we describe the main results in the paper (See Table 1 for a summary). When parameterized by the deletion distance k to cluster graphs,

- we can find a minimum dominating set in $\mathcal{O}^*(3^k)$ time. Within the same time, we can also find a minimum independent dominating set (IDS) or a minimum efficient dominating set (EDS) or a minimum total dominating set. We also give an $\mathcal{O}^*((r+2)^k)$ algorithm for minimum threshold dominating set with threshold r . These algorithms are obtained through a dynamic programming approach for interesting generalizations of set cover which may be of independent interest. These results are discussed in Sect. 4.1.
- We complement our upper bound results by showing that for dominating set and total dominating set, $\mathcal{O}^*((2-\epsilon)^k)$ algorithm is not possible for any $\epsilon > 0$ under what is known as Set Cover Conjecture. We also show that for IDS, $\mathcal{O}^*((2-\epsilon)^k)$ algorithm is not possible for any $\epsilon > 0$ under the Strong Exponential Time Hypothesis (SETH) and for EDS no $2^{o(k)}$ algorithm is possible unless the Exponential Time Hypothesis (ETH) is false. It also follows from our reductions that dominating set, TDS and IDS do not have polynomial sized kernels unless $\text{NP} \subseteq \text{coNP}/\text{poly}$. These results are discussed in Sect. 4.2.

The standard dominating set and most of its variants are NP-hard or W[2]-hard in split graphs [20]. For the two variants IDS and EDS that are polynomial time solvable in split graphs, we show that when parameterized by the deletion distance k to split graphs,

- IDS can be solved in $\mathcal{O}^*(2^k)$ time and provide an $\mathcal{O}^*((2-\epsilon)^k)$ lower bound for any $\epsilon > 0$ assuming SETH. We also show that IDS-SVD has no polynomial kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.
- The 2^k barrier can be broken for EDS by designing an $\mathcal{O}^*(3^{k/2})$ algorithm. This is one of the very few problems with a runtime better than $\mathcal{O}^*(2^k)$ in the realm of structural parameterization. We also show that no $2^{o(k)}$ algorithm is possible unless the ETH is false. These results are discussed in Sect. 5.

Table 1. Summary of results. Results marked \star indicate our results.

	Cluster deletion set		Split deletion set	
	Algorithms	Lower bounds	Algorithms	Lower bounds
DS, TDS	$\mathcal{O}^*(3^k) \star$	$\mathcal{O}^*((2 - \epsilon)^k)$ and No polynomial kernel \star		para-NP-hard
IDS	$\mathcal{O}^*(3^k) \star$	$\mathcal{O}^*((2 - \epsilon)^k)$ and No polynomial kernel	$\mathcal{O}(2^k) \star$	$\mathcal{O}^*((2 - \epsilon)^k)$ and No polynomial kernel
EDS	$\mathcal{O}^*(3^k) \star$	$\mathcal{O}^*(2^{o(k)}) \star$	$\mathcal{O}^*(3^{k/2}) \star$	$\mathcal{O}^*(2^{o(k)}) \star$
ThDS	$\mathcal{O}^*((r + 2)^k) \star$	No polynomial kernel \star		para-NP-hard

2 Preliminaries and Notations

We use $[n]$ to denote the set $\{1, \dots, n\}$. We use standard terminologies of graph theory book by Diestel [10]. For a graph $G = (V, E)$ we denote n as the number of vertices and m as the number of edges. For a vertex $v \in V(G)$, we denote $N_G(v) = \{(u \in V(G) | (u, v) \in E(G))\}$ as the open neighborhood of v . When there is no confusion, we drop the subscript G . By $N[v]$ we denote the close neighborhood of v , i.e. $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, we denote $N(S) = \{v \in V(G) | \exists u \in S \text{ such that } (u, v) \in E(G)\} \setminus S$. And we denote $N[S] = N(S) \cup S$. By $N^{=2}(v)$ we denote the set of vertices that are at minimum distance exactly two from v . For $S \subseteq V(G)$, we denote $G[S]$ to be the subgraph induced on S . We say that for vertices $u, v \in V$, u dominates v if $v \in N(u)$.

We give a general template of formal definition of problems as follows:

\mathcal{P} - \mathcal{Q}	Parameter: $ S $
Input: An undirected graph $G = (V, E)$, $S \subseteq V(G)$ which is a \mathcal{Q} and an integer ℓ .	
Question: Is there a \mathcal{P} in G of size atmost ℓ ?	

where \mathcal{P} represents an acronym of a dominating set variant among DS, EDS, IDS, TDS and ThDS and \mathcal{Q} that of a modulator among CVD and SVD. For example, in the EDS-SVD problem we are interested in finding an EDS of size atmost ℓ given a k sized SVD where k is the parameter.

We use the following conjectures and theorems to prove some of our lower bounds.

Conjecture 1 (Strong Exponential Time Hypothesis (SETH)) ([17]). There is no $\epsilon > 0$ such that $\forall q \geq 3$, q -CNFSAT can be solved in $\mathcal{O}^*((2 - \epsilon)^n)$ time where n is the number of variables in input formula.

Conjecture 2 (Exponential Time Hypothesis (ETH)) ([16, 17]). 3-CNF-SAT cannot be solved in $\mathcal{O}^*(2^{o(n)})$ time where the input formula has n variables and m clauses.

Conjecture 3 (Set Cover Conjecture (SCC)) ([7]). There is no $\epsilon > 0$ such that SET COVER can be solved in $\mathcal{O}^*((2 - \epsilon)^n)$ time where n is the size of the universe.

Theorem 1 ([11]). SET-COVER parameterized by the universe size does not admit any polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$.

Theorem 2 ([14]). CNF-SAT parameterized by the number of variables admits no polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$.

3 Related Work

Clique-width [6] of a graph is a parameter that measures how close to a clique the graph is. Courcelle et al. [5] showed that for a graph with clique-width at most k , any problem expressible in MSO_1 (monadic second order logic of the first kind) has an FPT algorithm with k as the parameter if a k -expression for the graph (a certificate showing that the clique-width of the graph is at most k) is also given as input. The clique-width of a graph that is k away from a cluster graph can be shown to be $k + 1$ (with a k -expression) and all the dominating set variants discussed in the paper can be expressed in MSO_1 and hence can be solved in FPT time in such graphs. But the running time function $f(k)$ in Courcelle's theorem is huge (more than doubly exponential). Oum et al. [19] gave an $\mathcal{O}^*(k^{O(k)})$ algorithm to solve the minimum dominating set for clique-width k graphs without assuming that the k -expression is given. There is a $\mathcal{O}^*(4^k)$ algorithm by Bodlaender et al. [2] for finding minimum dominating set in graphs with clique-width k when the k -expression is given as input. It is easy to construct the k -expression for graphs k away from a cluster graph and hence we have a $\mathcal{O}^*(4^k)$ algorithm. The algorithms we give in Sect. 4, not only improve the running time but also are applicable for other variants of dominating set.

4 Dominating Set Variants Parameterized by CVD Size

4.1 Upper Bounds

In cluster graphs, a dominating set simply picks an arbitrary vertex from each clique. This dominating set is also efficient and independent. For threshold dominating set with threshold r , we arbitrarily pick $r + 1$ vertices from every clique if possible so that every vertex has r neighbors excluding itself.

We can assume that the CVD set S of size k is given with the input. If not, we can use the algorithm by Boral et al. [3] that runs in $\mathcal{O}^*(1.92^k)$ time and either outputs a CVD set of size at most k or says that no such set exists.

We first look at the problem DS-CVD which is NP-hard as any graph having an edge has a CVD set of at most $n - 2$.

Theorem 3. DS-CVD can be solved in $\mathcal{O}^*(3^k)$ time.

Proof. Our FPT algorithm starts with making a guess S' for the solution's intersection with S . We delete vertices in $N[S'] \cap S$ as they have been already dominated. We will keep the vertices of $N[S'] \cap (V \setminus S)$ as they can be used to cover the remaining vertices of S .

Let us denote the cliques in $G' = G \setminus S$ as C_1, C_2, \dots, C_q where $q \leq n - k$. We label the vertices of G' as $v_1, v_2, \dots, v_{|V \setminus S|}$ such that the first l_1 of them belong to the clique C_1 , the next l_2 of them belong to clique C_2 and so on for integers l_1, l_2, \dots, l_q . Note that for some cliques, all the vertices of the clique gets dominated by S' . We are left with the problem of picking the minimum number of vertices from the cliques to dominate, the vertices of the cliques that are not yet dominated (by S'), and $S \setminus N[S']$. We abstract out the problem below.

DS-DISJOINTCLUSTER

Parameter: $|S|$

Input: An undirected graph $G = (V, E)$, $S \subseteq V$ such that every connected component of $G \setminus S$ is a clique, a $(0, 1)$ vector (f_1, f_2, \dots, f_q) corresponding for the cliques (C_1, \dots, C_q) and an integer ℓ .

Question: Does there exist a subset $T \subseteq V \setminus S$ of size ℓ , that dominates all vertices of S and all vertices of all cliques C_i with flags $f_i = 1$?

For the problem we started off with, the set S in this new formulation is the remaining vertices of S after deleting $N[S'] \cap S$. Also f_i is set to 1 if the clique C_i has not been dominated by S' and is set to 0 otherwise.

Lemma 1. DS-DISJOINTCLUSTER can be solved in $\mathcal{O}^*(2^{|S|})$ time.

Proof. We formulate this problem instance as a variant of SET-COVER instance. Define the universe U as the set S . For each vertex $v \in V \setminus S$, we define a set $S_v = N(v) \cap S$. Define the family of sets $\mathcal{F} = \{S_v | v \in V \setminus S\}$. We say that a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ covers a subset $W \subseteq U$ if for every element $w \in W$, there exist some set in \mathcal{F}' containing w . Now a SET-COVER solution $\mathcal{F}' \subseteq \mathcal{F}$ for (U, \mathcal{F}) will cover all the elements of S . In the graph, the vertices corresponding to the sets in \mathcal{F}' will dominate all the vertices in S . But DS-DISJOINTCLUSTER has the additional requirement of dominating the vertices of every clique C_i with $f_i = 1$ as well. This means from every such clique at least one vertex has to be picked. With this in mind, we define for each clique C_i a collection of sets $\mathcal{B}_i = \{S_v : v \in C_i\}$. We call these sets as *blocks*. Hence the number of blocks and the number of cliques in $G \setminus S$ are the same. We order the sets in the block in the order of the vertices $v_1, \dots, v_{|V \setminus S|}$. We have the following problem which is a slight generalization of SET-COVER.

SET-COVER WITH PARTITION

Parameter: $|U| = k$

Input: A universe U , a family of sets $\mathcal{F} = \{S_1, \dots, S_m\}$, a partition $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_q)$ of \mathcal{F} , a $(0, 1)$ vector (f_1, f_2, \dots, f_q) corresponding to each block in the partition $(\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_q)$ and an integer ℓ .

Question: Does there exist a subset $\mathcal{F}' \subseteq \mathcal{F}$ of size ℓ covering U and from each block \mathcal{B}_i with flags $f_i = 1$ at least one set is picked?

Lemma 2 (\star) .¹ SET-COVER WITH PARTITION can be solved in $\mathcal{O}^*(2^{|U|})$ time.

We construct the SET-COVER WITH PARTITION instance from the DS-DISJOINTCLUSTER instance as discussed above. It can be easily seen that there exists a solution of size ℓ in DS-DISJOINTCLUSTER instance if and only if there exists a solution of size ℓ in SET-COVER WITH PARTITION instance. In Lemma 2, we solve SET-COVER WITH PARTITION via dynamic programming. And $|S| = |U|$. This completes the proof. \square

Now for each guess $S' \subseteq S$ with $|S'| = i$, we construct the SET-COVER WITH PARTITION instance with $|U| \leq k - i$ and solve it with running time $\mathcal{O}^*(2^{k-i})$. Hence the total running time is $\sum_{i=1}^k \binom{k}{i} \mathcal{O}^*(2^{k-i})$ which is $\mathcal{O}(3^k n^{\mathcal{O}(1)})$. \square

We show that with some careful modifications to the above dynamic programming algorithm, efficient FPT algorithms for minimum EDS, IDS, TDS and ThDS when parameterized by the size of cluster deletion set can be obtained.

Theorem 4 (\star) . EDS-CVD, TDS-CVD and IDS-CVD can be solved in $\mathcal{O}^*(3^k)$ time. THDS-CVD can be solved in $\mathcal{O}^*((r+2)^k)$ time.

4.2 Lower Bounds

Lemma 3 (\star) . There is a polynomial time algorithm that takes an instance (U, \mathcal{F}, ℓ) of SET-COVER and outputs an instance (G, ℓ) of DS-CVD (or TDS-CVD) such that G has a cluster vertex deletion set with exactly $|U|$ vertices, such that (U, \mathcal{F}, ℓ) has a set cover of size ℓ if and only if G has a (total) dominating set of size ℓ .

The following theorem follows from the above lemma and Conjecture 3.

Theorem 5 (\star) . DS-CVD and TDS-CVD cannot be solved in $\mathcal{O}^*((2-\epsilon)^k)$ running time for any $\epsilon > 0$ unless Set Cover Conjecture fails.

The following theorem follows from Theorem 1 and Lemma 3.

Theorem 6 (\star) . DS-CVD, TDS-CVD and THDS-CVD do not have polynomial sized kernels unless $\text{NP} \subseteq \text{coNP/poly}$.

Note that the proof idea of Theorem 6 does not work for IDS-CVD. To show $\mathcal{O}^*((2-\epsilon)^k)$ lower bound for IDS-CVD under SETH, we use the following theorem and an observation. Here MMVC-VC problem refers to the problem of finding a maximum sized minimal vertex cover (MMVC) in a graph parameterized by the size of a given vertex cover (VC). Recall that a vertex cover in a graph is a subset of vertices that covers all edges.

¹ Due to lack of space, the proofs of Theorems, Lemmas, Observations, Safeness of Reduction Rules marked \star and some omitted details will appear in the full version.

Theorem 7 ([21]).² *Unless SETH fails, MMVC-VC cannot be solved in $\mathcal{O}^*((2-\epsilon)^k)$ time. Moreover, MMVC-VC does not admit polynomial sized kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.*

Observation 1 (\star). *If T is a minimal vertex cover of the graph G , then $V(G)\setminus T$ is an independent dominating set in G . Furthermore, if T is a maximum minimal vertex cover, then $V(G)\setminus T$ is a minimum independent dominating set.*

From Observation 1, we know that the complement of a maximum minimal vertex cover is a minimum independent dominating set. Also, any vertex cover is a cluster vertex deletion set. So, from Theorem 7, we have the following result.

Corollary 1 (\star). *IDS-CVD cannot be solved in $\mathcal{O}^*((2-\epsilon)^k)$ time for any $\epsilon > 0$ unless SETH fails. Moreover, IDS-CVD does not have any polynomial kernel unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.*

For EDS-CVD, we can only prove a weaker lower bound of $2^{o(k)}$ time assuming ETH, but we give the lower bound for EDS parameterized by even a larger parameter, i.e. the size of a vertex cover. We have the following results.

Theorem 8 (\star). *EDS-VC cannot be solved in $2^{o(|S|)}$ time unless ETH fails.*

Corollary 2. *EDS-CVD cannot be solved in $2^{o(|S|)}$ time unless ETH fails.*

5 Dominating Set Variants Parameterized by SVD Size

In this section, we address the parameterized complexity of dominating set variants when parameterized by the size of a given SVD set S . Note that DS and TDS are NP-hard on split graphs [20]. Hence we focus only on EDS and IDS.

We assume that S is given with the input. Otherwise given (G, k) , we use an $\mathcal{O}^*(1.27^{k+o(k)})$ algorithm due to Cygan and Pilipczuk [9] to find a set of vertices of size at most k whose removal makes G into a split graph.

5.1 EDS and IDS Parameterized by SVD Size

First, we provide a simple algorithm for IDS-SVD. The idea is to make a guess for the solution within the SVD and solve the resulting disjoint problem in polynomial time. It turns out that it works for EDS-SVD too.

Theorem 9 (\star). *EDS-SVD and IDS-SVD can be solved in $\mathcal{O}^*(2^k)$ time.*

² Note that the SETH based lower bound result and the result ruling out the existence of polynomial kernel in this paper use different constructions.

5.2 Lower Bounds for IDS and EDS

We know that any vertex cover is a split vertex deletion set. So, we have the following corollary as a consequence of Theorem 7.

Corollary 3 (\star).³ *IDS-SVD cannot be solved in $\mathcal{O}^*((2 - \epsilon)^k)$ time unless SETH fails and it does not admit polynomial kernels unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.*

For EDS, as the size of the SVD set is always smaller than the size of the vertex cover, we have the following corollary of Theorem 8.

Corollary 4. *EDS-SVD cannot be solved in $2^{o(|S|)}$ time unless ETH fails.*

5.3 Improved Algorithm for EDS-SVD

In this section, we give an improved algorithm for EDS-SVD parameterized by the size of a given split vertex deletion set S breaking the barrier of $\mathcal{O}^*(2^k)$. Let $F = G \setminus S$. As F is a split graph, $V(F) = C \uplus I$ where C induces a clique and I induces an independent set. The algorithm uses the standard branching technique. Consider any efficient dominating set D of a graph. Any two vertices $u, v \in D$ must have distance at least three. At any intermediate stage of the algorithm, we make a choice of not picking a vertex and we mark such vertices by coloring them red. Other vertices are colored blue. Hence all vertices of G are blue initially. We initialize $D = \emptyset$ which is the solution set we seek. Consider any pair of blue vertices $x, y \in S$. If the distance between x and y is at most two in G , then we use the following branching rule. And we measure the progress of the algorithm by $\mu(G)$ which is the number of blue vertices in S , which is k initially.

Branching Rule 1. *Consider a pair of blue vertices $x, y \in S$ such that the distance between x and y is at most two in G . In the first branch, we add x into D , delete $N[x]$ from G , color the vertices in $N^{=2}(x)$ by red. In the second branch, we add y into D , delete $N[y]$ from G , color the vertices in $N^{=2}(y)$ by red. In the third branch, we color x, y by red.*

Clearly the branches are exhaustive as both x and y cannot be in the EDS solution we seek. Furthermore, in the first branch, x is deleted from S and y is colored red. Symmetrically in the second branch, y is deleted from S and x is colored red. In the third branch, x and y are colored red. So in all the branches, $\mu(G)$ drops by at least two resulting in a $(2, 2, 2)$ branching rule. When this branching rule is not applicable, for every pair of blue vertices $x, y \in S$, $N[x] \cap N[y] = \emptyset$. Now, as C is a clique, we can have at most one vertex from C in the solution. When we decide to pick some vertex $v \in C$ into the solution, then we delete $N[v]$ and color $N^{=2}(v)$ as red. So all vertices of C get deleted. There are at most $|C|$ vertices in C . When we decide not to pick any vertex from C into the solution, then we color all vertices of C as red. So we have $(|C| + 1)$

³ We provide an alternate proof in the full version.

choices from the vertices of C . Measure $\mu(G)$ does not increase in any of these choices. A multiplicative factor of $(|C| + 1)$ would come in the running time because of this one-time branching. Now, we are left with only the vertices of I . Now, we apply the following reduction rule to rule out some simple boundary conditions.

Reduction Rule 1. *If there exists a red vertex $x \in V(G)$ such that $N_G(x)$ has only one blue vertex y , then add y into D , delete $N[y]$ from G and color $N^{=2}(y)$ as red. Also if there exists a blue vertex $x \in V(G)$ such that $N_G(x)$ contains no blue vertex, then add x into D , delete $N[x]$ from G and color $N^{=2}(x)$ as red.*

It is easy to see that the above reduction rule is safe. Note that we have some blue vertices in I . Such vertices can only be dominated by themselves or a unique blue vertex in S , as otherwise Branching Rule 1 would have been applicable. Now, suppose that there exists a blue vertex $x \in S$ that has at least two blue neighbors $u, v \in I$. If we decide to pick u (or symmetrically v) into D , then we are not allowed to pick x or v (symmetrically u) in D but then u or v cannot be dominated. This forces x into D . We have the following reduction rule.

Reduction Rule 2 (\star). *If there exists a blue vertex $x \in S$ such that $N_G(x)$ contains at least two blue neighbors in I , then add x into D , delete $N[x]$ from G and color vertices in $N^{=2}(x)$ red.*

Lemma 4 (\star). *Reduction Rules 1 and 2 do not increase $\mu(G)$.*

Now if there are red vertices in I having no blue neighbor in S , then we move to the next branch as such a vertex cannot be dominated. Thus any blue vertex in I has only one blue neighbor in S and any blue vertex in S has only one blue neighbor in I . As Reduction Rule 1 is not applicable, any red vertex $x \in S \cup C$ has at least two blue neighbors in $u, v \in N_G(x)$. Clearly both $\{u, v\} \not\subset S$ as otherwise Branching Rule 1 would have been applicable. So, now we are left with the case that $u, v \in I$ or $u \in I, v \in S$ but (u, v) may or may not be an edge. Now we apply the following branching rule.

Branching Rule 2. *Let x be a red vertex in S with two blue neighbors u, v .*

1. *If $u, v \in I$, then we branch as follows. In one branch we add u into D , delete $N[u]$ from G , color $N^{=2}(u)$ as red. As $v \in N^{=2}(u)$ and v has only one blue neighbor $z \in S$, we add z also into D , delete $N[z]$ from G and color $N^{=2}(z)$ by red. In the second branch, we add v into D , delete $N[v]$ from G , color $N^{=2}(v)$ as red. As $u \in N^{=2}(v)$ and u has only one blue neighbor $y \in S$, we add y also into D , delete $N^{=2}(y)$ from G and color $N^{=2}(z)$ by red. In the third branch, color both u and v by red. Add the only blue neighbor y of u and z of v into D . Delete $N[y], N[z]$ from G and color the vertices in $N^{=2}(y) \cup N^{=2}(z)$ by red.*
2. *$u \in I, v \in S, (u, v) \notin E(G)$, then we branch as follows. In the first branch, we add u to D , color v as red. This forces us to pick the only blue neighbor z of v where $z \in I$. So, we add z to D . Delete $N[u], N[z]$ from G and color*

$N^{=2}(u), N^{=2}(z)$ as red. In the second branch, e color u as red. This forces us to pick the only neighbor y of u where $y \in S$. And we pick v into D as well as y into D . We delete $N[v], N[y]$ from G and color $N^{=2}(v), N^{=2}(y)$ by red. In the third branch, we color both u and v by red. This forces us to pick the only blue neighbor $z \in N_G(v) \cap I, y \in N_G(u) \cap S$ into D . So, we pick z into D , delete $N[z], N[y]$ from G and color $N^{=2}(y), N^{=2}(z)$ by red.

It is easy to see that $\mu(G)$ drops by at least two in all three branches as eventually two blue vertices of S get deleted in all the branches.

When none of the above rules are applicable, then we have $u \in S, v \in I$ and $(u, v) \in E(G)$. We know that either $u \in D$ or $v \in D$. Consider the red vertices in $N(u)$ and red vertices in $N(v)$. As Branching Rule 1, Reduction Rule 1 and Branching Rule 2 are not applicable, by the following lemma using which we can pick u or v arbitrarily.

Lemma 5 (\star). *If Branching Rule 1, Reduction Rule 1 and Branching Rule 2 are not applicable, then $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.*

This completes the description of our algorithm that consists of a sequence of reduction rules and branching rules. The measure is k initially and the branching continues as long as k drops to 0. So, we have the following recurrence.

$$T(k) \leq 3T(k-2) + \alpha \cdot (n+k)^c$$

Solving this recurrence, we get $\mathcal{O}(1.732^k \cdot n^{\mathcal{O}(1)})$ implying the following theorem.

Theorem 10. *EDS-SVD can be solved in $\mathcal{O}^*(3^{k/2})$ time.*

6 Concluding Remarks

We have initiated a study of structural parameterizations of some dominating set variants and complemented with lower bounds based on ETH and SETH. One immediate open problem is to narrow the gap between upper and lower bounds, especially for the dominating set variants parameterized by the size of CVD set.

We know that IDS is the complementary version of MAXIMUM MINIMAL VERTEX COVER problem. So a natural approach for an $\mathcal{O}^*(2^k)$ algorithm for IDS-CVD is to apply the ideas used in [21] to get $\mathcal{O}^*(2^k)$ algorithm for MMVC-VC. But this seems to require more work as there may not exist a minimal vertex cover that intersects the CVD set S in a particular subset.

Recently Bergougnoux and Kanté [1] have given an $\mathcal{O}^*(2^{\mathcal{O}(k)})$ algorithm for connected dominating set (the dominating set induces a connected graph) for clique-width k graphs when the k -expression is given as input. An interesting open problem is whether connected dominating set has a simpler FPT algorithm as in the FPT algorithms in this paper, when parameterized by the CVD set size.

References

1. Bergougnoux, B., Kanté, M.M.: Fast exact algorithms for some connectivity problems parametrized by clique-width. arXiv preprint [arXiv:1707.03584](https://arxiv.org/abs/1707.03584) (2017)
2. Bodlaender, H.L., van Leeuwen, E.J., van Rooij, J.M.M., Vatshelle, M.: Faster algorithms on branch and clique decompositions. In: Hliněný, P., Kučera, A. (eds.) MFCS 2010. LNCS, vol. 6281, pp. 174–185. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15155-2_17
3. Boral, A., Cygan, M., Kociumaka, T., Pilipczuk, M.: A fast branching algorithm for cluster vertex deletion. *Theory Comput. Syst.* **58**(2), 357–376 (2016)
4. Cai, L.: Parameterized complexity of vertex colouring. *Discrete Appl. Math.* **127**(3), 415–429 (2003)
5. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.* **33**, 125–150 (2000)
6. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. *Discrete Appl. Math.* **101**(1), 77–114 (2000)
7. Cygan, M., Dell, H., Lokshtanov, D., Marx, D., Nederlof, J., Okamoto, Y., Paturi, R., Saurabh, S., Wahlström, M.: On problems as hard as CNF-SAT. *ACM Trans. Algorithms (TALG)* **12**(3), 41 (2016)
8. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-319-21275-3>
9. Cygan, M., Pilipczuk, M.: Split vertex deletion meets vertex cover: new fixed-parameter and exact exponential-time algorithms. *Inf. Process. Lett.* **113**(5–6), 179–182 (2013)
10. Diestel, R.: *Graph Theory*. Springer, Heidelberg (2006)
11. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and IDs. *ACM Trans. Algorithms (TALG)* **11**(2), 13 (2014)
12. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness II: on completeness for $W[1]$. *Theor. Comput. Sci.* **141**(1–2), 109–131 (1995)
13. Fellows, M.R., Jansen, B.M.P., Rosamond, F.A.: Towards fully multivariate algorithmics: parameter ecology and the deconstruction of computational complexity. *Eur. J. Comb.* **34**(3), 541–566 (2013)
14. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.* **77**(1), 91–106 (2011)
15. Haynes, T.W., Hedetniemi, S., Slater, P.: *Domination in Graphs: Advanced Topics*. Marcel Dekker, New York (1997)
16. Impagliazzo, R., Paturi, R.: On the complexity of k -SAT. *J. Comput. Syst. Sci.* **62**, 367–375 (2001)
17. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.* **63**(4), 512–530 (2001)
18. Jansen, B.M.P., Raman, V., Vatshelle, M.: Parameter ecology for feedback vertex set. *Tsinghua Sci. Technol.* **19**(4), 387–409 (2014)
19. Oum, S.-I., Sæther, S.H., Vatshelle, M.: Faster algorithms parameterized by clique-width. arXiv preprint [arXiv:1311.0224](https://arxiv.org/abs/1311.0224) (2013)
20. Raman, V., Saurabh, S.: Short cycles make W -hard problems hard: FPT algorithms for W -hard problems in graphs with no short cycles. *Algorithmica* **52**(2), 203–225 (2008)
21. Zehavi, M.: Maximum minimal vertex cover parameterized by vertex cover. *SIAM J. Discrete Math.* **31**(4), 2440–2456 (2017)