On the Parameterized Complexity of Minus Domination

Sriram Bhyravarapu[1(✉)], Lawqueen Kanesh[2(✉)], A Mohanapriya[3(✉)], Nidhi Purohit[4(✉)], N. Sadagopan[3(✉)], and Saket Saurabh[1,4(✉)]

[1] The Institute of Mathematical Sciences, HBNI, Chennai, India
{sriramb,saket}@imsc.res.in

[2] Indian Institute of Technology, Jodhpur, Jodhpur, India
lawqueen@iitj.ac.in

[3] Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram, Chennai, India
{coe19d003,sadagopan}@iiitdm.ac.in

[4] University of Bergen, Bergen, Norway
Nidhi.Purohit@uib.no

Abstract. Dominating Set is a well-studied combinatorial problem. Given a graph $G = (V, E)$, a dominating function $f : V(G) \to \{0, 1\}$ is a labeling of the vertices of G such that $\sum_{w \in N[v]} f(w) \geq 1$ for each vertex $v \in V(G)$, where $N[v] = \{v\} \cup \{u \mid uv \in E(G)\}$. We study a generalization of Dominating Set called Minus Domination (in short, MD) where $f : V(G) \to \{-1, 0, 1\}$. Such a function is said to be a *minus dominating function* if for each vertex $v \in V(G)$, we have $\sum_{w \in N[v]} f(w) \geq 1$. The objective is to minimize the weight of a minus domination function, which is $f(V) = \sum_{u \in V(G)} f(u)$. The problem is NP-hard even on bipartite, planar, and chordal graphs.

In this paper, we study MD from the perspective of parameterized complexity. After observing the complexity of the problem with the natural parameters such as the number of vertices labeled 1, -1 and 0, we study the problem with respect to structural parameters. We show that MD is fixed-parameter tractable when parameterized by twin-cover number, neighborhood diversity or the combined parameters component vertex deletion set and size of the largest component. In addition, we give an XP-algorithm when parameterized by distance to cluster number.

Keywords: Minus Domination · fixed-parameter tractability · twin-cover · neighborhood diversity · disjoint paths deletion · cluster vertex deletion

1 Introduction

Given a graph $G = (V, E)$, a *dominating function* $f : V(G) \to \{0, 1\}$ is a labeling of $V(G)$ from $\{0, 1\}$ such that for each vertex $v \in V(G)$ we have $\sum_{w \in N[v]} f(w) \geq 1$, where $N[v] = \{v\} \cup \{u \mid uv \in E(G)\}$. The weight of f is denoted by $f(V) = \sum_{u \in V(G)} f(u)$. The Dominating Set (in short, DS)

H. Fernau et al. (Eds.): SOFSEM 2024, LNCS 14519, pp. 96–110, 2024.
https://doi.org/10.1007/978-3-031-52113-3_7

problem asks to find a dominating function of minimum weight. Several variants of DS have been studied in literature, some of which include independent, total, global, perfect and k-dominating [1,12,13]. In this paper, we study another variant of domination called MINUS DOMINATION (in short, MD) which was introduced by Dunbar et al. in 1996 [6]. Given a graph $G = (V, E)$, a *minus dominating function* $f : V(G) \rightarrow \{-1, 0, 1\}$ is an assignment of labels to the vertices of G such that for each $v \in V(G)$, the sum of labels assigned to the vertices in the closed neighborhood of v (denoted by $N[v]$) is at least one, i.e., $\sum_{w \in N[v]} f(w) \geq 1$. The *weight* of a minus dominating function f denoted by $f(V)$ is $\sum_{v \in V(G)} f(v)$. Given a graph G, MINUS DOMINATION asks to compute the minimum weight of a minus dominating function of G. The decision version of the problem takes as input a graph G and an integer k, and outputs whether there exists a minus dominating function of weight at most k. MD has applications in electrical networks, social networks, voting, etc. [6,19].

The weight of a minus dominating function can be negative. For example, consider a clique on n vertices and for each edge uv in the clique, add a private vertex adjacent to only u and v. Consider the minus dominating function f that assigns all the clique vertices the label 1 and all the private vertices the label -1. The clique vertices have as many private neighbors as they have clique neighbors while the private vertices have exactly two clique neighbors. Thus, for each vertex v, we have $f(N[v]) \geq 1$ and $f(V) = n - n(n-1)/2 < 0$ for a large n. The authors in [6] show that given a positive integer k there exists a bipartite, chordal and outer-planar graphs with weight at most $-k$.

MINUS DOMINATION is NP-complete in general [8] and NP-complete even on chordal bipartite graphs, split graphs, and bipartite planar graphs of degree at most 4 [3,6,8,17]. The problem is polynomial-time solvable on trees, graphs of bounded rank-width, cographs, distance hereditary graphs and strongly chordal graphs [6,8]. Given the hardness results for the problem, it is natural to ask for ways to confront this hardness. Parameterized complexity is an approach towards solving NP-hard problems in "feasible" time. Parameterized problems that admit such an algorithm are called fixed-parameter tractable (in short, FPT). For more details, we refer the reader to the book by Cygan et al. [2] and Downey and Fellows [4].

MINUS DOMINATION has been studied from the realm of parameterized complexity. On subcubic graphs, MD is FPT when parameterized by weight [18]. As far as near-optimal solutions are concerned, the minimum weight of a minus dominating function cannot be approximated in polynomial time within $(1 + \epsilon)$, for some $\epsilon > 0$, unless $\mathsf{P} \neq \mathsf{NP}$ [3]. The problem is APX-hard on graphs of maximum degree 7 [18]. Several combinatorial bounds for the problem on regular graphs, and small degree graphs ($\Delta \leq 3$ or $\Delta \leq 4$) have been studied [3,7].

A parameter may originate from the formulation of the problem itself (called natural parameters) or it can be a property of the input graph (called structural parameters). DOMINATING SET when parameterized by solution size is W[2]-hard [4]; however, when parameterized by structural parameters such as tree-width [2], modular-width, or distance to cluster (size of the cluster vertex deletion set) [11], the problem is fixed-parameter tractable. MD has various natural parameters such as n_{-1}, n_0, n_1 and $f(V)$ (where $n_{-1} = |f^{-1}(-1)|$, $n_0 = |f^{-1}(0)|$, and $n_1 = |f^{-1}(1)|$ for a minus dominating function f) and it was shown in [8] that the problem is para-NP-hard when parameterized by $f(V)$.

Domination vs Minus Domination: One may think that the ideas used for solving DS can be extended to MD. But this is not the case always. There are graphs for example connected cographs, wheel graphs, windmill graphs, chain graphs, etc. where dominating set size is constant and the corresponding set can be found trivially. However, it is not the case with MD.

On graphs of bounded tree-width, the dynamic programming based FPT algorithm for DS when parameterized by tree-width (tw) focuses on guessing vertices from each bag that are in the dominating set. However, for MD, just guessing the labels 1, −1 and 0 for the vertices of a bag does not suffice. We may also need to store the information about the sum that each vertex in the bag receives from its subtree to be able to extend to the rest of the graph. Since the degree of a vertex can be unbounded, the sum it receives from the subtree can be unbounded. This gives us an $n^{\mathcal{O}(\mathsf{tw})}$ time algorithm. Notice that this gives us an FPT algorithm when parameterized by maximum degree and tree-width. The authors of [8] believe that MD is not FPT when parameterized by tree-width or rank-width. To the best of our knowledge, the FPT status of minus domination with respect to tree-width is still open.

Our Contribution: First, we analyse the problem on natural parameters. We obtain the following result, the proof of which is not hard and follows from Dominating Set and its well-known variants.

Theorem 1 (⋆).[1] Minus Domination *when parameterized by* n_{-1} *or* n_0 *is para-*NP*-hard, when parameterized by* n_1 *or* $n_{-1} + n_1$ *is* W[2]*-hard, and when parameterized by* $n_0 + n_1$ *is FPT.*

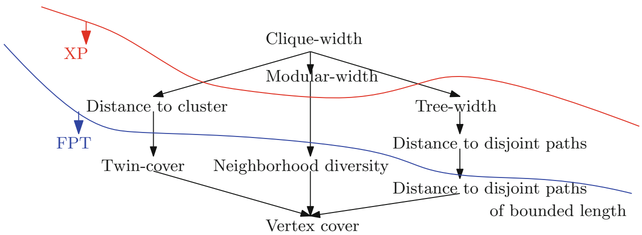

Fig. 1. Hasse diagram of graph parameters for MD. A directed edge from the parameter a to the parameter b indicates that $a \leq g(b)$ for some computable function g. The parameters below the blue curve are those for which MD is FPT while the parameters between the red and blue curves are those for which XP algorithms are known for MD. (Color figure online)

[1] Due to space constraints, all the proofs of the results marked (⋆) will be presented in the full version of the paper.

We shift our focus to various structural graph parameters. In Sect. 3, we show that MD is FPT when parameterized by twin-cover number. The next parameter to consider is distance to cluster number, which is a generalization of twin-cover number. In Sect. 4, we obtain an XP-algorithm when parameterized by distance to cluster number. Then we move our attention to a more general parameter: the size of component vertex deletion set. In Sect. 5, we study the problem on this parameter and obtain an FPT algorithm when parameterized by the size of component vertex deletion set and the size of a largest component. This implies an FPT algorithm for MD when parameterized by (i) distance to cluster number and the size of a largest clique, (ii) distance to disjoint paths and the size of a largest path, or (iii) feedback vertex set number and the size of a largest tree component. We also show that MD is FPT when parameterized by the parameter neighborhood diversity. An illustration of the results is given in Fig. 1. We now state the theorems of the above discussed results.

Theorem 2. Minus Domination *can be solved in* $2^{\mathcal{O}(k \cdot 2^k)} n^{\mathcal{O}(1)}$ *time, where* k *is the twin cover number of the graph.*

Theorem 3. Minus Domination *can be solved in time* $g(k) \cdot n^{2k+6}$*, where* k *is the distance to cluster number.*

Theorem 4 (⋆). *Let* G *be a graph and* $S \subseteq V(G)$ *of size* k *be such that* $G - S$ *is a disjoint union of components where each component has at most* d *vertices. Then,* Minus Domination *is FPT when parameterized by* k *and* d*.*

Theorem 5 (⋆). Minus Domination *can be solved in time* $t^{\mathcal{O}(t)} n^{\mathcal{O}(1)}$*, where* t *is the neighborhood diversity of the graph.*

Open Question: What is the parameterized complexity of MD when parameterized by distance to cluster, tree-width or feedback vertex set, or distance to disjoint paths?

2 Preliminaries

In this paper, we consider finite, undirected and connected graphs. If the graph is disconnected, then we apply our algorithms on each of the components independently. Given a graph $G = (V, E)$, we use $V(G)$ and $E(G)$ to denote the vertex and the edge sets of G. For a vertex $v \in V(G)$, we use $N(v)$ (open neighborhood of v) to denote the neighbors of v in G. The closed neighborhood of v is denoted by $N[v] = N(v) \cup \{v\}$. For a vertex $v \in V(G)$ and a set $C \subseteq V(G)$, we denote $N_C(v) = N(v) \cap C$. For a pair of vertices $u, v \in V(G)$, we say u and v are *true twins*, if and only if $N[u] = N[v]$. We say that a vertex v *satisfies the sum property*, if $\sum_{u \in N[v]} f(u) \geq 1$. For a set $X \subseteq V(G)$ and a vertex w, we say w *receives the sum* s from X if $\sum_{v \in N[w] \cap X} f(v) = s$.

The *size* of a minus dominating function is the number of vertices assigned the label 1. For a nonempty subset $S \subseteq V(G)$, we denote by $G[S]$ the subgraph of G induced by S. Let $f : X \to Y$ be a function. If $A \subseteq X$ then the restriction of f to A is the function $f|_A : A \to Y$ given by $f|_A(x) = f(x)$, for $x \in A$. We say a labeling $f : X \to \{-1, 0, 1\}$ *extends* $g : Y \to \{-1, 0, 1\}$ if $Y \subseteq X$ and for each $w \in Y$, we have $f(w) = g(w)$. We use $\mathcal{O}^*$ notation to hide factors that are polynomial in the input size.

Definition 1 (Twin-cover [10]). *Given a graph G, a set $S \subseteq V(G)$ is called a* twin cover *of G if the following conditions hold: (i) $G[V \backslash S]$ is a disjoint union of cliques, and (ii) each pair of vertices of a clique in $G[V \backslash S]$ are true twins in G. We then say that G has twin cover number k if k is the minimum possible size of a twin cover of G.*

Definition 2 (Distance to cluster [14]). *A* cluster graph *is a disjoint union of cliques. Given a graph G, a set of vertices $S \subseteq V(G)$ is called a* cluster vertex deletion set *of G if $G - S$ is a cluster graph. The size of the smallest set S for which $G - S$ is a cluster graph is referred to as* distance to cluster number*.*

Definition 3 (Neighborhood diversity [16]). *Let $G = (V, E)$ be a graph. Two vertices $u, v \in V(G)$ are said to have the* same type *if and only if $N(u)\backslash\{v\} = N(v)\backslash\{u\}$. A graph G has neighborhood diversity at most t, if there exists a partition of $V(G)$ into at most t sets $V_1, V_2, \dots, V_t$ such that all the vertices in each set have the same type.*

Ganian [10] showed that a twin-cover of size k can be found in time $\mathcal{O}^*(1.2738^k)$. Hüffner et al. [14] showed that showed that a cluster vertex deletion set of size k can be computed in $\mathcal{O}^*(1.811^k)$ time. Lampis [16] showed that the neighborhood diversity of a graph can be found in polynomial time. Thus we will assume that a twin-cover, a cluster vertex deletion set and a partition of vertex set into types of vertices are given as input, in the respective sections.

We use Integer Linear Programming (ILP) feasibility problem, stated in [15] as subroutine in some of our results.

Theorem 6 ([15]). *An integer linear programming instance of size L with p variables can be solved using*

$$\mathcal{O}(p^{2.5p+o(p)} \cdot (L + \log M_x) \log(M_x M_c))$$

arithmetic operations and space polynomial in $L + \log M_x$, where M_x is an upper bound on the absolute value a variable can take in a solution, and M_c is the largest absolute value of a coefficient in the vector c.

Lemma 1. *Let $f : V(G) \to \{-1, 0, 1\}$ be a minus dominating function and $u, v \in V(G)$ be true twins such that $f(u) = 1$ and $f(v) = -1$. Then there exists a minus dominating function $g : V(G) \to \{-1, 0, 1\}$ of weight $f(V)$ and $g(u) = g(v) = 0$.*

Proof. We construct a function $g : V(G) \to \{-1, 0, 1\}$ as follows: $g(u) = g(v) = 0$ and $g(z) = f(z)$, for each $z \in V(G)\backslash\{u, v\}$. We claim that g is the minus dominating function of weight $f(V)$. It is easy to see that $g(V) = f(V)$ because $g(u) + g(v) = f(u) + f(v) = 0$ and the remaining vertices are assigned the same labels in both the labelings. Now we show that g is a minus dominating function. It is easy to see that for each vertex $w \in V(G)$ that is not adjacent to either u or v, $\sum_{y \in N[w]} g(y) = \sum_{y \in N[w]} f(y) \geq 1$, as u and v are the only vertices whose labels are changed. Since u and v are true twins, for each $w \in N(u) \cup N(v)$,

$$\sum_{y \in N[w]} g(y) = g(u) + g(v) + \sum_{y \in N[w] \setminus \{u,v\}} g(y) = f(u) + f(v) + \sum_{y \in N[w] \setminus \{u,v\}} f(y)$$
$$= \sum_{y \in N[w]} f(y) \geq 1.$$

Thus, g is a minus dominating function. □

3 Twin-Cover

Let G be a graph and $S \subseteq V(G)$ be a twin cover of G of size k. Let $C_1, C_2, \ldots, C_\ell$ be the set of maximal cliques in $G - S$. From the definition of twin cover, we have that each vertex of a clique C in $G[V \setminus S]$ has the same neighborhood in S. We denote the neighborhood of a clique C in S by $N_S(C)$, i.e., $N_S(C) = N(C) \cap S$. We partition the cliques in $G[V \setminus S]$ based on its neighborhood in S. For each non-empty subset $A \subseteq S$, we denote by $T_A = \{C \mid C \text{ is a maximal clique in } G[V \setminus S] \text{ and } N_S(C) = A \subseteq S\}$ the set of cliques where each clique is adjacent to every vertex in A. We call T_A a *clique type*. Notice that the number of clique types is at most 2^k. Consider the case when $A = \emptyset$. If $T_A \neq \emptyset$ then G is disconnected. Since we are considering connected graphs, we can assume that $A \neq \emptyset$ throughout this section. In addition, we only consider $A \subseteq S$ for which $|T_A| \geq 1$, or T_A is non-empty.

Next, we show that for each $A \subseteq S$, the vertices of cliques in set T_A receive their labeling from a fixed set of labels.

Lemma 2 (⋆). *Let $f : V(G) \to \{-1, 0, 1\}$ be any minus dominating function with minimum weight. Then there exists a minus dominating function $g : V(G) \to \{-1, 0, 1\}$ with weight $f(V)$ such that for each $A \subseteq S$, vertices in T_A receives labels from exactly one of the three sets $\{-1, 0\}, \{0, 1\}$ or $\{0\}$.*

Proof (Proof of Theorem 2). Let G be a graph and $S \subseteq V(G)$ be a twin cover of G of size k. Let $f : V(G) \to \{-1, 0, 1\}$ be a minus dominating function of minimum weight. WLOG we can assume that f is a minus dominating function such that for each $A \subseteq S$, the vertices in T_A receive the labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$, as any minus dominating function can be converted to the function satisfying the above said conditions using Lemma 2. Recall that there are $2^k - 1$ many clique types.

For each $A \subseteq S$, we denote $b_A \in \{\{-1, 0\}, \{0, 1\}, \{0\}\}$ to be the set of labels that the vertices of the cliques in T_A are labeled with. We say

$\widehat{b} = \{b_A : A$ is non-empty and $|T_A| \geq 1\}$ *respects* f if for each non-empty $A \subseteq S$ with $|T_A| \geq 1$, the vertices of the cliques in T_A are assigned the labels from b_A.

We guess the tuple $(f|_S, \widehat{b}, \widehat{s})$, where $\widehat{s} = f(V) = \sum_{v \in V(G)} f(v)$ is the minimum weight and $-n < \widehat{s} \leq n$. For each guess, we formulate the problem as an ILP feasibility problem. Let $c_A = \sum_{v \in A} f|_S(v)$ denote the sum that each vertex in the clique type T_A receives from $A \subseteq S$. Let ℓ_A represent the number of cliques in each clique type T_A and $d_A = \min\{|D| : D$ is a clique in $T_A\}$ represent the minimum size of a clique in T_A. Let m_A denote the total number of vertices over all the cliques in the clique type T_A.

From now on, we work with a fixed guess $(f|_S, \widehat{b}, \widehat{s})$. Observe that for each $A \subseteq S$, c_A, ℓ_A, d_A and m_A are constants. We say a guess $(f|_S, \widehat{b}, \widehat{s})$ is *invalid* if for any $A \subseteq S$, one of the following holds: (i) $c_A \leq 1$ and $b_A = \{-1, 0\}$ (i.e., the vertices of T_A are assigned the labels from $\{-1, 0\}$), (ii) $c_A \leq 0$ and $b_A = \{0\}$, and (iii) $b_A = \{0, 1\}$ and $c_A \leq -d_A$. Otherwise, we call $(f|_S, \widehat{b}, \widehat{s})$ a *valid* guess.

For a valid guess $(f|_S, \widehat{b}, \widehat{s})$, we formulate an instance of the ILP problem. The goal of ILP is to obtain an assignment of the variables such that each vertex $v \in V(G)$ satisfies its sum property respecting $\widehat{b}$ and $\widehat{s}$. For each $A \subseteq S$, let $n_{-1,A}$, $n_{0,A}$ and $n_{1,A}$ be the variables of the ILP instance that denote the number of vertices assigned -1, 0 and 1 respectively in the clique type T_A. Notice that the number of variables is at most $3 \cdot 2^k$. We now describe the constraints of ILP for each $A \subseteq S$ as follows.

(C1) The number of vertices from T_A assigned the labels from $\{-1, 0, 1\}$ is m_A. That is,

$$n_{-1,A} + n_{0,A} + n_{1,A} = m_A.$$

(C2) If $b_A = \{0\}$ and $c_A \geq 1$, then

$$n_{-1,A} = n_{1,A} = 0, \text{ and } n_{0,A} = m_A.$$

(C3) If $b_A = \{0, 1\}$ and $c_A > -d_A$, then

$$n_{-1,A} = 0, n_{0,A} \geq 0 \text{ and } (1 - c_A)\ell_A \leq n_{1,A} \leq m_A.$$

(C4) If $b_A = \{-1, 0\}$ and $c_A > 1$, then

$$n_{1,A} = 0, n_{0,A} \geq 0, \text{ and}$$

$$0 \leq n_{-1,A} \leq \sum_{j=1}^{\ell_A} \min\{c_A - 1, |B_j|\}.$$

where $B_1, B_2, \ldots, B_{\ell_A}$ are the cliques of the type T_A.

(C5) For each $v \in S$, the sum property is satisfied. That is,

$$\sum_{A: v \in A} (n_{1,A} - n_{-1,A}) + \sum_{w: w \in N[v] \cap S} f(w) \geq 1.$$

(C6) Weight of our desired minus dominating function is $\widehat{s}$,

$$\sum_{A} n_{1,A} - n_{-1,A} + \sum_{v \in S} f(v) = \widehat{s}.$$

We next show a one-to-one correspondence between feasible assignments of ILP and minus dominating functions of G.

Lemma 3. *ILP has a feasible assignment if and only if there exists a minus dominating function with weight* $\widehat{s}$.

Proof. Suppose that there is a feasible assignment returned by ILP. We show that there exists a minus dominating function $f : V(G) \rightarrow \{-1, 0, 1\}$ respecting $f|_S$, $\widehat{b}$ and $\widehat{s}$. For each $A \subseteq S$, we assign the labels to the vertices of the clique type T_A in the following manner. Let $V(T_A)$ be the vertices of the cliques in T_A.

- **Case 1:** $b_A = \{0\}$.
  Assign $f(v) = 0$ for each vertex v in the cliques of T_A.
- **Case 2:** $b_A = \{0, 1\}$ and $c_A > -d_A$.
  We have the following subcases.
  - $c_A \geq 1$.
    Choose $n_{1,A}$ vertices arbitrarily from the cliques of T_A and assign the label 1 to them. Rest of the vertices (if any) are assigned the label 0.
  - $c_A \leq 0$.
    Choose $(1 - c_A)$ vertices arbitrarily from each clique of T_A and assign the label 1 to them. From the constraint (C3), we have that $n_{1,A} \geq (1 - c_A)\ell_A$. The remaining $n_{1,A} - (1 - c_A)\ell_A$ vertices are picked arbitrarily from the unassigned vertices of the cliques of T_A.
- **Case 3:** $b_A = \{-1, 0\}$ and $c_A \geq 2$.
  For each clique $B_j \in T_A$, choose $\min\{c_A - 1, |B_j|\}$ many vertices and assign the label -1 to each of them. Remaining vertices (if any) are assigned the label 0.

Clearly, the labeling $f : V(G) \rightarrow \{-1, 0, 1\}$ respects $\widehat{b}$, $\widehat{s}$ and the fact that the vertices in each clique type T_A receive the set of labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$ from the constraints (C1), (C2), (C3), (C4) and (C6). The sum property for each vertex $v \in S$ is satisfied because of Constraint (C5). Therefore it is sufficient to show that for each $v \in V(G)$, the sum property is satisfied.

Consider a vertex v from T_A. If T_A falls under Case 1, then v receives its positive sum from A from constraint (C2). If T_A falls under Case 2 and $c_A \geq 1$, then v receives its positive sum from A irrespective of whether v is assigned 0 or 1, because of constraint (C3). Else if T_A falls under Case 2 and $c_A \leq 0$, then we ensured that each clique in T_A has at least $(1 - c_A)$ vertices assigned the label 1, from constraint (C3), making the total sum in neighborhood of v to be at least 1. If T_A falls under Case 3, then we ensured that the number of vertices assigned -1 in each clique B_j of T_A is $\min\{c_A - 1, |B_j|\}$ from constraint (C4), making the closed neighborhood sum to be at least 1 for v. Notice that the above argument works irrespective of the label assigned to v.

Conversely, let $f : V(G) \rightarrow \{-1, 0, 1\}$ be a minus dominating function respecting $f|_S$, $\widehat{b}$ and $\widehat{s}$. Thus the constraint (C6) is satisfied. The variables are assigned as follows depending on the labeling of T_A.

- $b_A = \{0\}$.
  Assign $n_{0,A} = m_A$ and $n_{-1,A} = n_{1,A} = 0$.
- $b_A = \{0, 1\}$ and $c_A \geq -d_A$.
  Assign $n_{-1,A} = 0$, $n_{0,A} = f^{-1}(0) \cap V(T_A)$ and $n_{1,A} = f^{-1}(1) \cap V(T_A)$.
- $b_A = \{-1, 0\}$ and $c_A \geq 2$.
  Assign $n_{1,A} = 0$, $n_{0,A} = f^{-1}(0) \cap V(T_A)$ and $n_{-1,A} = f^{-1}(-1) \cap V(T_A)$.

Each vertex from T_A is assigned a label from $\{-1, 0, 1\}$ and hence constraint (C1) is satisfied. The assignment of labels to vertices in $V \backslash S$ is such that every vertex in V satisfies sum property. Thus the constraints (C2), (C3) and (C4) are satisfied. Each vertex in S satisfied the sum property in f. Thus the constraint (C5) is satisfied. □

We run ILP over all the valid guesses and check whether there exists an assignment leading to a minus dominating function. Over all such assignments we pick the assignment that has the minimum $\widehat{s}$.

Running Time: Guessing a labeling of S, the set of labels the vertices in each T_A can receive, and the weight $\widehat{s}$, takes time $\mathcal{O}(3^k \cdot 3^{2^k} \cdot n)$. For each of the above guesses, we run the ILP feasibility problem where the number of variables is at most $3 \cdot 2^k$. Thus from Lemma 3 and Theorem 6, the total time taken is $2^{\mathcal{O}(k \cdot 2^k)} \cdot n^{\mathcal{O}(1)}$. □

4 Cluster Vertex Deletion Set

Let G be a graph and $S \subseteq V(G)$ be a cluster vertex deletion set of size k. Also let $C_1, C_2, \ldots, C_\ell$ be the maximal cliques of $G - S$. We partition the vertices of each clique C_i, $i \in [\ell]$, based on its neighborhood in S. For each $A \subseteq S$, we use $C_{i,A} = \{v \mid v \in C_i \text{ and } N(v) \cap S = A\}$ to denote the set of vertices from C_i that are adjacent to each vertex in A. Next, we show that for each clique C_i, the vertices in $C_{i,A}$ receive their labels from a fixed set of labels, for each $A \subseteq S$. Notice that A could be an empty set.

Lemma 4. *Let $f : V(G) \rightarrow \{-1, 0, 1\}$ be a minus dominating function. Then there exists a minus dominating function of weight $f(V)$ such that in each clique C_i, for each $A \subseteq S$ and a non-empty $C_{i,A}$, the vertices of $C_{i,A}$ receive labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$.*

Proof. If all vertices in $C_{i,A}$ are assigned the label 0 then the claim is trivially satisfied. For each $i \in \{1, \ldots, \ell\}$ and $A \subseteq S$, if f assigns the vertices of $C_{i,A}$ from the labels $\{1, 0\}$ or $\{-1, 0\}$, then we conclude that f is the desired function.

Otherwise, there exists a clique C_i and an $A \subseteq S$ with vertices u and v in $C_{i,A}$ such that $f(u) = 1$ and $f(v) = -1$ or $f(u) = -1$ and $f(v) = 1$. WLOG let $f(u) = 1$ and $f(v) = -1$ (similar arguments apply for the other case). Since u and v are true twins, we apply Lemma 1 and obtain a minus dominating function of weight $f(V)$ with u and v assigned the label 0.

After repeated application of Lemma 1 on each $C_{i,A}$, where $A \subseteq S$ and $i \in [t]$, all vertices in $C_{i,A}$ are either assigned labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$. □

From now on, we can assume that in any minus dominating function, for each $A \subseteq S$ and a clique C_i, the vertices in $C_{i,A}$ are assigned labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$.

We now look at the following lemma. Suppose that we are given a labeling of S, the number of vertices in a clique C_i assigned the labels -1 and 1, and the sum each vertex in S receives from C_i. Then we can decide whether there exists a assignment of labels to C_i extending the labeling of S and satisfying the assumptions.

Lemma 5. *Let $f : S \to \{-1, 0, 1\}$ be a labeling of S and C_i be a clique in $G - S$. Let $a_i, b_i \in \mathbb{N} \cup \{0\}$. Also let $X^i = (x^i_1, \ldots, x^i_k)$ be a tuple where x^i_j corresponds to $v_j \in S$. Then there is an algorithm that runs in $\mathcal{O}^*(2^{O(k \cdot 2^k)})$ and either returns a labeling $g : S \cup C_i \to \{-1, 0, 1\}$ that extends f with the following properties,*

- *$a_i = |g^{-1}(1) \cap C_i|$, $b_i = |g^{-1}(-1) \cap C_i|$,*
- *$\forall v_j \in S$, $x^i_j = \sum_{u \in N_{C_i}(v_j)} g(u)$,*
- *for each $A \subseteq S$ and a non-empty $C_{i,A}$, the vertices of $C_{i,A}$ receive the labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$, and*
- *for each $v \in C_i$, $\sum_{w \in N_{C_i}[v]} g(w) \geq 1$,*

or returns that there is no labeling g extending f satisfying the properties.

Proof. Given a labeling $f : S \to \{-1, 0, 1\}$, a clique C_i, $X^i = (x^i_1, \ldots, x^i_k)$, and two integers a_i and b_i, the goal is to find a labeling $g : S \cup C_i \to \{-1, 0, 1\}$ extending f satisfying some constraints. We formulate this as an ILP feasibility problem with the variables: $n_{1,A}$, $n_{-1,A}$ and $n_{0,A}$ that denote the number of vertices in $C_{i,A}$, for each $A \subseteq S$, that are assigned the labels -1, 0 and 1 respectively. The number of variables is at most $3 \cdot 2^k$. We now present the constraints.

(C1) For each $A \subseteq S$, the number of vertices assigned the labels -1, 0, and 1 in $C_{i,A}$ is at least 0 and at most $|C_{i,A}|$. In addition, each vertex is assigned some label.

$$0 \leq n_{-1,A}, n_{1,A}, n_{0,A} \leq |C_{i,A}| \text{ and } n_{-1,A} + n_{1,A} + n_{0,A} = |C_{i,A}|.$$

(C2) For each $v_j \in S$, the sum it receives from C_i is x^i_j.

$$\sum_{A:\ v_j \in A} n_{1,A} - n_{-1,A} = x^i_j.$$

(C3) For each $A \subseteq S$, the vertices in $C_{i,A}$ are assigned the labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$.

$$n_{1,A} > 0 \implies n_{-1,A} = 0,$$
$$n_{-1,A} > 0 \implies n_{1,A} = 0.$$

(C4) Total sum of vertices assigned the labels 1 and -1 are a_i and b_i respectively.

$$\sum_A n_{1,A} = a_i, \text{ and } \sum_A n_{-1,A} = b_i.$$

(C5) For each $A \subseteq S$, the sum property for vertices in $C_{i,A}$ is satisfied.

$$\sum_{A\colon v \in A} f(v) + a_i - b_i \geq 1.$$

We now have to show that there is a feasible assignment of ILP if and only if there is a labeling g that extends f and satisfying the properties.

Feasibility Implies Labeling: Let there be a feasible assignment of values to variables returned by the ILP. In each $C_{i,A}$, choose $n_{1,A}$, $n_{0,A}$ and $n_{-1,A}$ many vertices arbitrarily and assign them the label 1, 0 and -1, respectively. Notice that each vertex is assigned a label because of Constraint (C1). For each $v_j \in S$, the sum it receives from C_i is x^i_j, which is ensured from Constraint (C2). For each $A \subseteq S$, the vertices in $C_{i,A}$ are assigned labels from exactly one of the three sets $\{-1, 0\}$, $\{0, 1\}$ or $\{0\}$ which is ensured by Constraint (C3) and Constraint (C1). The number of vertices assigned 1 and -1 is a_i and b_i respectively and is ensured by Constraint (C4). Every vertex in C_i satisfies the sum property and this is ensured by Constraint (C5).

Labeling Implies Feasibility: Let $g : S \cup C_i \to \{-1, 0, 1\}$ be an extension of f satisfying the properties of a_i, b_i, x^i_j, the set of labels used in each $C_{i,A}$ and the sum property for each vertex in C_i with respect to g. Using Lemma 4, we convert g to be a labeling such that each $C_{i,A}$ receives the labels from exactly one of the three sets $\{-1, 0\}$, $\{1, 0\}$ or $\{0\}$. We obtain a feasible assignment for variables in ILP as follows. For each $A \subseteq S$, we set $n_{-1,A} = |g^{-1}(-1) \cap C_{i,A}|$, $n_{0,A} = |g^{-1}(0) \cap C_{i,A}|$, and $n_{1,A} = |g^{-1}(1) \cap C_{i,A}|$. By definition of g, the constraints (C1), (C2), (C3), (C4), and (C5) are satisfied.

Since the number of variables is at most $3 \cdot 2^k$, from Theorem 6, the running time of our algorithm is $2^{\mathcal{O}(k \cdot 2^k)} n^{\mathcal{O}(1)}$. □

We say a tuple (a_i, b_i, X^i) is *feasible* for C_i if Lemma 5 returns a feasible labeling of C_i extending the labeling of S. Else we call it infeasible. We now proceed to the proof of Theorem 3.

Proof (Proof of Theorem 3). Let G be a graph and $S \subseteq V(G)$ of size k be such that $G - S$ is a disjoint union of cliques. Let $C_1, C_2, \ldots, C_\ell$ be the cliques of $G - S$. Let $g : V(G) \to \{-1, 0, 1\}$ be a minus dominating function of minimum weight. The first step of the algorithm is to guess the labeling $g|_S$ of S. For each clique C_i in $G - S$, we try to obtain a labeling of C_i (if one exists) that extends $g|_S$ using Lemma 5. Towards this, we guess the following: a_i and b_i, which are the number of vertices assigned the labels 1 and -1 respectively in C_i, and $X^i = (x^i_1, \ldots, x^i_k)$ the tuple where x^i_p corresponds to the sum that the vertex $v_p \in S$ receives from C_i. Thus for each of the guesses, we should able to decide whether there exists a labeling of $C_i \cup S$ extending $g|_S$ that satisfies the sum property for each vertex in C_i.

We give a bottom-up dynamic programming based algorithm to find g.

An entry $T[i, a, b, X]$ in the table is set to 1 if there exists a labeling of the vertices in $C_1, C_2, \cdots, C_i$ such that

- a is the number of vertices labelled 1 over the cliques from C_1 to C_i,
- b is the number of vertices labelled -1 over the cliques from C_1 to C_i,
- $X = (x_1, x_2, \cdots, x_k)$ is the tuple where x_j corresponds to the sum the vertex $v_j \in S$ receives from the cliques C_1 to C_i, and
- each vertex in $C_1 \cup C_2 \cup \cdots \cup C_i$ satisfies the sum property.

Otherwise, we store $T[i, a, b, X] = 0$.

We now define the recurrence. For an entry $T[i, a, b, X]$, we go over all feasible tuples of C_i and look at the corresponding subproblem over the cliques C_1 to C_{i-1}. That is,

$$T[i, a, b, X] = \bigvee_{\text{feasible tuples } (a_i, b_i, X^i) \text{ of } C_i} T[i-1, a-a_i, b-b_i, (x_1 - x^i_1, \ldots, x_k - x^i_k)].$$

Notice that in a feasible tuple we have $n \geq a \geq a_i \geq 0$, $n \geq b \geq b_i \geq 0$ and $n \geq x_j \geq x^i_j \geq -n$ for all j. The base case of the recurrence is obtained at the clique C_1, which is computed as follows:

$$T[1, a, b, X] = \begin{cases} 1 & (a, b, X) \text{ is a feasible tuple for } C_1, \\ 0 & \text{otherwise.} \end{cases}$$

The correctness of the algorithm follows from the description. We now compute the running time of the algorithm. The number of labelings $g|_S$ is at most 3^k and the number of feasible tuples for a clique C_i is at most n^{k+2}. Using Lemma 5, we can decide if a tuple is feasible or not in time $2^{\mathcal{O}(k \cdot 2^k)} n^{\mathcal{O}(1)}$. The number of entries in T is at most n^{k+3}. For each of the entry in T, we go over all feasible tuples of C_i and thus the total running time is $2^{\mathcal{O}(k \cdot 2^k)} n^{2k+6}$. □

5 Distance to Disjoint Components and Component Size

Let G be a graph and $k, d \in \mathbb{N}$ be two integers. We consider the problem of computing a set $S \subseteq V(G)$ of size at most k such that $G - S$ is a disjoint union of connected components where each connected component has at most d vertices. This problem is known in literature as d-Component Order Connectivity (in short, d-COC) [9]. Notice that when $d = 1$, d-COC is the Vertex Cover problem. There is a $\mathcal{O}(\log d)$-approximation algorithm for d-COC [5,9].

In this section, we consider MD when parameterized by k and d where the input is a graph G, an integer ℓ and a set $S \subseteq V(G)$ of size k such that $G - S$ is a disjoint union of components each of size at most d vertices. The objective is to check whether there exists a minus dominating function of weight at most ℓ.

Towards this, we consider the solution set obtained from the approximation algorithm of d-COC as our modulator set S. Notice that $|S| \leq \mathcal{O}(k \log d)$, if a solution exists for d-COC. We now provide a proof sketch of Theorem 4.

Proof Sketch of Theorem 4. Let $\mathcal{C} = \{C \mid C \text{ is a component in } G-S\}$ denote the set of components in $G-S$. Notice that for each $C \in \mathcal{C}$, we have $1 \leq |C| \leq d$. Let $V(C) = \{u_1, u_2, \ldots, u_{d'}\}$, $d' \leq d$. For each component $C \in \mathcal{C}$, we apply the following procedure.

- We find the *equivalence class* of C based on its neighborhood in $S \cup V(C)$. Let $\mathcal{T} = \{T_A \mid A \subseteq S \cup V(C)\}$. Since $|S| = k$ and $|C| \leq d$, we have $|\mathcal{T}| = 2^{k+d}$. An equivalence class is defined by the function $g : \mathcal{C} \to \mathcal{T}^d$. Note that the number of equivalence classes is at most $2^{(k+d)d}$.
  The equivalence class $g(C)$ is denoted by $(T_{A_1}, T_{A_2}, \ldots, T_{A_{|d'|}})$ where for each $u_i \in V(C)$ we have $N(u_i) \cap S = A_i$.
- We now consider all possible labelings $h : V(C) \to \{-1, 0, 1\}$ of C. We say a labeling h is *feasible* for C, when the vertices of C are assigned the labels from h and each vertex v of C satisfies the sum property (i.e., the sum that v receives from $C \cup S$ is at least one).
  The set of feasible labelings $\mathcal{H}_{g(C)} = \{h \mid h \text{ is feasible for } C \text{ and } C \text{ belongs to the equivalence class } g(C)\}$ is constructed. Notice that all the components that belong to an equivalence class $g(C)$ have the same set of feasible labelings.
- We formulate an ILP feasibility instance using the above information. The variables of ILP represent the number of components in $G - S$ belonging to an equivalence class that receive a particular labeling from the feasible list of labelings (of that equivalence class). The number of variables for each equivalence class $g(C)$ is equal to $|\mathcal{H}_{g(C)}|$.

The running time of the algorithm is majorly dependent on the running time of the ILP which in turn depends on the number of variables which is $2^{(k+d)d}3^d$. $\square$

Using Theorem 4, we get the following results when the components are cliques and trees respectively.

Corollary 1. MD *is FPT when parameterized by (i) cluster vertex deletion number and size of a largest clique, or (ii) feedback vertex set number and size of a largest tree.*

Acknowledgement. We would like to thank the anonymous reviewers for their helpful comments. The first author acknowledges SERB-DST for supporting this research via grant PDF/2021/003452. The fifth author acknowledges NBHM for supporting this research via project NBHM-02011/24/2023/6051. The fifth author would also like to acknowledge DST for supporting this research via project CRG/2023/007127.

References

1. Chang, G.J.: Algorithmic aspects of domination in graphs. In: Pardalos, P.M., Du, D.-Z., Graham, R.L. (eds.) Handbook of Combinatorial Optimization, pp. 221–282. Springer, New York (2013). https://doi.org/10.1007/978-1-4419-7997-1_26
2. Cygan, M., et al.: Parameterized Algorithms. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
3. Damaschke, P.: Minus domination in small-degree graphs. Discrete Appl. Math. **108**(1), 53–64 (2001). Workshop on Graph Theoretic Concepts in Computer Science
4. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer, London (2013). https://doi.org/10.1007/978-1-4471-5559-1
5. Drange, P.G., Dregi, M., van 't Hof, P.: On the computational complexity of vertex integrity and component order connectivity. Algorithmica **76**(4), 1181–1202 (2016). https://doi.org/10.1007/s00453-016-0127-x
6. Dunbar, J., Goddard, W., Hedetniemi, S., McRae, A., Henning, M.A.: The algorithmic complexity of minus domination in graphs. Discrete Appl. Math. **68**(1), 73–84 (1996)
7. Dunbar, J., Hedetniemi, S., Henning, M.A., McRae, A.A.: Minus domination in regular graphs. Discrete Math. **149**(1), 311–312 (1996)
8. Faria, L., Hon, W.-K., Kloks, T., Liu, H.-H., Wang, T.-M., Wang, Y.-L.: On complexities of minus domination. Discrete Optim. **22**, 6–19 (2016). SI: ISCO 2014
9. Fomin, F.V., Gaspers, S., Kratsch, D., Liedloff, M., Saurabh, S.: Iterative compression and exact algorithms. Theor. Comput. Sci. **411**(7), 1045–1053 (2010)
10. Ganian, R.: Twin-cover: beyond vertex cover in parameterized algorithmics. In: Marx, D., Rossmanith, P. (eds.) IPEC 2011. LNCS, vol. 7112, pp. 259–271. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28050-4_21
11. Goyal, D., Jacob, A., Kumar, K., Majumdar, D., Raman, V.: Structural parameterizations of dominating set variants. In: Fomin, F.V., Podolskii, V.V. (eds.) CSR 2018. LNCS, vol. 10846, pp. 157–168. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-90530-3_14
12. Haynes, T.W., Hedetniemi, S., Slater, P.: Fundamentals of Domination in Graphs. CRC Press, Boca Raton (1998)
13. Hedetniemi, S.T., Laskar, R.C.: Bibliography on domination in graphs and some basic definitions of domination parameters. Discrete Math. **86**(1), 257–277 (1990)
14. Hüffner, F., Komusiewicz, C., Moser, H., Niedermeier, R.: Fixed-parameter algorithms for cluster vertex deletion. Theory Comput. Syst. **47**(1), 196–217 (2010). https://doi.org/10.1007/s00224-008-9150-x
15. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
16. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. Algorithmica **64**(1), 19–37 (2012). https://doi.org/10.1007/s00453-011-9554-x

17. Lee, C.-M., Chang, M.-S.: Variations of Y-dominating functions on graphs. Discret. Math. **308**(18), 4185–4204 (2008)
18. Lin, J.-Y., Liu, C.-H., Poon, S.-H.: Algorithmic aspect of minus domination on small-degree graphs. In: Xu, D., Du, D., Du, D. (eds.) COCOON 2015. LNCS, vol. 9198, pp. 337–348. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21398-9_27
19. Zheng, Y., Wang, J., Feng, Q.: Kernelization and lower bounds of the signed domination problem. In: Fellows, M., Tan, X., Zhu, B. (eds.) AAIM/FAW -2013. LNCS, vol. 7924, pp. 261–271. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38756-2_27