

MIXED DOMINATING SET: A Parameterized Perspective

Pallavi Jain¹, M. Jayakrishnan¹, Fahad Panolan², and Abhishek Sahu¹(✉)

¹ The Institute of Mathematical Sciences, HBNI, Chennai, India
{pallavi.j, jayakrishnan, asahu}@imsc.res.in

² Department of Informatics, University of Bergen, Bergen, Norway
fahad.panolan@ii.uib.no

Abstract. In the MIXED DOMINATING SET (MDS) problem, we are given an n -vertex graph G and a positive integer k , and the objective is to decide whether there exists a set $S \subseteq V(G) \cup E(G)$ of cardinality at most k such that every element $x \in (V(G) \cup E(G)) \setminus S$ is either adjacent to or incident with an element of S . We show that MDS can be solved in time $7.465^k n^{\mathcal{O}(1)}$ on general graphs, and in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ on planar graphs. We complement this result by showing that MDS does not admit an algorithm with running time $2^{o(k)} n^{\mathcal{O}(1)}$ unless the Exponential Time Hypothesis (ETH) fails, and that it does not admit a polynomial kernel unless $\text{coNP} \subseteq \text{NP/poly}$. In addition, we provide an algorithm which, given a graph G together with a tree decomposition of width tw , solves MDS in time $6^{\text{tw}} n^{\mathcal{O}(1)}$. We finally show that unless the Set Cover Conjecture (SeCoCo) fails, MDS does not admit an algorithm with running time $\mathcal{O}((2 - \epsilon)^{\text{tw}(G)} n^{\mathcal{O}(1)})$ for any $\epsilon > 0$, where $\text{tw}(G)$ is the tree-width of G .

1 Introduction

Dominating (or covering) objects such as vertices and edges in a graph by vertices or edges give rise to several classic problems, such as VERTEX COVER, EDGE COVER, DOMINATING SET and EDGE DOMINATING SET (see Table 1). All these problems and their numerous variants have been studied extensively from structural as well as algorithmic points of view. However, all these problems except EDGE COVER are known to be NP-complete [11, 25], and thus, they have been subjected to intense scrutiny in all the algorithmic paradigms meant for coping with NP-hardness, including approximation algorithms and parameterized complexity. In this paper we consider a well-studied variant of these problems, where the objective is to dominate *vertices and edges* by *vertices and edges*.

In order to define the problems formally, we first define the notion of domination, that is, what a vertex or an edge can dominate. A *vertex dominates* itself, all its neighbors and all the edges incident with it. On the other hand, an

The research leading to these results have received funding from the European Research Council via ERC Advanced Investigator Grant 267959.

Table 1. Different domination problems and their FPT and kernelization status.

Dominating	By	Problem	PC	Poly kernel
Vertices	Vertices	DOMINATING SET	W[2]-hard	No
Vertices	Edges	EDGE COVER	P	$\mathcal{O}(1)$
Edges	Edges	EDGE DOMINATING SET	FPT	Yes
Edges	Vertices	VERTEX COVER	FPT	Yes
Edges+vertices	Vertices	VERTEX COVER	FPT	Yes
Edges+vertices	Edges	EDGE COVER	P	$\mathcal{O}(1)$
Edges+vertices	Edges+vertices	MIXED DOMINATING SET	FPT	No

edge dominates its two endpoints, and all the edges incident with either of its endpoints. We first define the problem of dominating vertices by vertices. A *dominating set* of a graph G is a set $S \subseteq V(G)$ such that every vertex $v \in V(G) \setminus S$ is adjacent to at least one vertex in S . In the DOMINATING SET problem, we are given an input graph G , a positive integer k , and the objective is to check whether there exists a dominating set of size at most k . The edge counterpart of DOMINATING SET is called EDGE DOMINATING SET. The problem we study in this paper is a variant of these domination problems. Towards that we first define the notion of *mixed dominating set (mds)*. Given a graph G , and a set $X \subseteq V(G) \cup E(G)$, X is called a *mds* if every element $x \in (V(G) \cup E(G)) \setminus X$ is either adjacent to or incident with an element of X . More formally, we study the following problem in the parameterized complexity framework.

MIXED DOMINATING SET (MDS)

Parameter: k or $\text{tw}(G)$

Input: A graph G on n vertices and m edges and a positive integer k .

Question: Does there exist a mds of size at most k in G ?

The notion of mds (also called total cover) was introduced in the 70s by Alavi et al. [1] as a generalization of matching and covering, and after that it has been studied extensively in graph theory [2, 9, 20, 22]. See the chapter in [14] for a survey on mds. The algorithmic complexity of MDS was first considered by Majumdar [18], where he showed that the problem is NP-complete on general graphs and admits a linear-time algorithm on trees. Hedetniemi et al. [15] and Manlove [19] showed that MDS remains NP-complete on bipartite and chordal graphs and on planar bipartite graphs of maximum degree 4, respectively. A decade and half later, Zhao et al. [26] considered MDS and showed that it remains NP-complete on split graphs. Unaware of the older result, they also designed an $\mathcal{O}(n \log n)$ time algorithm on trees. Lan and Chang [17] extended this result and gave a linear time algorithm for MDS on cacti (an undirected graph where any two cycles have at most one vertex in common). Hatami [13] gave a factor 2 approximation algorithm for MDS on general graphs. Recently,

Rajaati et al. [23] studied MDS parameterized by the treewidth of the input graph and designed an algorithm with running time $\mathcal{O}^*(3^{\text{tw}(G)^2})$.¹

In this paper we undertake a thorough study of MDS with respect to two parameters, the solution size and the treewidth of the input graph, and obtain the following results.

1. MDS admits an algorithm with running time $\mathcal{O}^*(7.465^k)$. We complement the FPT result by observing that (a) MDS does not admit an algorithm with running time $2^{o(k)}n^{\mathcal{O}(1)}$ unless ETH [16] fails; and (b) it does not admit a polynomial kernel unless $\text{coNP} \subseteq \text{NP/poly}$. See the last row of Table 1.
2. We design an algorithm with running time $\mathcal{O}^*(6^{\text{tw}(G)})$ for MDS parameterized by $\text{tw}(G)$. This algorithm is a significant improvement over the $\mathcal{O}^*(3^{\text{tw}(G)^2})$ algorithm of Rajaati et al. [23]. We also show that it does not admit an algorithm with running time $\mathcal{O}^*((2 - \epsilon)^{\text{tw}(G)})$, for any $\epsilon > 0$, unless SeCoCo fails [7].

The algorithm for MDS, parameterized by the solution size, is based on a relationship between mds and *vertex cover* (a subset X of vertices such that every edge has at least one endpoint in X) of the input graph. We use this connection to gain structural insights into the problem and give an algorithmically useful characterization of an optimal solution. This characterization leads us to the following algorithm: enumerate all the minimal vertex covers, say C , of size at most $2k$ of the input graph, guess a subset of C and then solve an appropriate auxiliary problem in polynomial time. The algorithm parameterized by treewidth uses standard dynamic programming approach together with subroutines for fast computation of cover-product [3]. Both our hardness results (no polynomial kernel and the lower bound on the running time of MDS parameterized by treewidth) are based on a polynomial time parameter preserving transformation from an appropriate parameterization of RED BLUE DOMINATING SET [8]. For references to algorithms and hardness mentioned in Table 1, and for an introduction to parameterized complexity, we refer to [6].

2 Preliminaries

All graphs in this paper are undirected and simple. For a graph G , $V(G)$ and $E(G)$ denote the set of vertices and edges of G , respectively. An edge between u and v in a graph G is represented by uv . For a set of edges $E' \subseteq E(G)$, we denote by $V(E')$, the set of vertices that are endpoints of edges in E' . For $v \in V(G)$, $N_G(v)$ denotes the set of neighbors of v , and $N_G[v] = N_G(v) \cup \{v\}$. Similarly, for a subset $V' \subseteq V(G)$, $N_G(V') = (\cup_{v \in V'} N_G(v)) \setminus V'$ and $N_G[V'] = N_G(V') \cup V'$. Also, for $V' \subseteq V(G)$, we denote by $G[V]$, the subgraph of G induced on V' . For a graph G and $R \subseteq V(G)$, we use $E(R)$ to denote the set of edges incident with at least one vertex in R . In this paper, V' shall always be a set vertices and E' , a

¹ \mathcal{O}^* notation suppresses the polynomial factor. That is, $\mathcal{O}(f(k)n^{\mathcal{O}(1)}) = \mathcal{O}^*(f(k))$.

set of edges, unless otherwise specified. We use \uplus to denote the disjoint union of two sets, i.e., for sets S, A and B , $S = A \uplus B$ means $S = A \cup B$ and $A \cap B = \emptyset$.

Treewidth. Let G be a graph. A *tree-decomposition* of a graph G is a pair $(\mathbb{T}, \mathcal{X} = \{X_t\}_{t \in V(\mathbb{T})})$ such that

- $\bigcup_{t \in V(\mathbb{T})} X_t = V(G)$,
- for all $xy \in E(G)$ there is a $t \in V(\mathbb{T})$ such that $\{x, y\} \subseteq X_t$, and
- for all $v \in V(G)$ the subgraph of \mathbb{T} induced by $\{t \mid v \in X_t\}$ is connected.

The *width* of a tree decomposition is $\max_{t \in V(\mathbb{T})} |X_t| - 1$ and the *treewidth* of G is the minimum width over all tree decompositions of G and is denoted by $\text{tw}(G)$.

3 Algorithm for MDS Parameterized by the Solution Size

In this section we design an algorithm for MDS parameterized by the solution size. We start with a simple observation that vertices and endpoints of edges in a mds form a vertex cover.

Lemma 1. *Let G be a graph and $S = V' \cup E'$ be a mds of G . Then $V' \cup V(E')$ is a vertex cover of G , of cardinality at most $2|S|$.*

Proof. Since $S = V' \cup E'$ is a mds of G , where $V' \subseteq V(G)$ and $E' \subseteq E(G)$, every edge in G has at least one of its endpoints in $V' \cup V(E')$. This implies that $V' \cup V(E')$ is a vertex cover of G , of cardinality at most $2|S|$. \square

In order to get a handle on an optimal solution we define what we call a *nice mds*.

*Among all **minimum sized** mixed dominating sets of G , pick the one with **the least number of vertices**. Such a mds is called a **nice mds**. To re-emphasize, a nice mds by definition is minimum sized.*

We now prove the following lemma, which forms the crux of our algorithm.

Lemma 2. *Let G be a connected graph and $V' \cup E'$ be a nice mds of G . Then, there is a minimal vertex cover C of G such that $V' \subseteq C \subseteq V' \cup V(E')$.*

Proof. Let $S = V' \cup E'$. Since S is mds, by Lemma 1, $V' \cup V(E')$ is a vertex cover of size at most $2|S|$. Any edge incident on $v \in V'$ dominates v as well as all the edges incident on v . Therefore, if v is such that $S \setminus \{v\}$ dominates $N_G(v)$, then by replacing v in S with some edge incident on v (this is possible since G is connected), we get another minimum sized mds with fewer vertices. This implies every vertex in V' must dominate at least one vertex (other than itself) which no other element in $V' \cup E'$ dominates. More specifically, for every $v \in V'$, there is a vertex $v' \in V(G)$ such that $vv' \in E(G)$ and $v' \notin N_G[(V' \setminus \{v\}) \cup V(E')]$. This means, every minimal vertex cover contained in $V' \cup V(E')$ must contain V' , because if $C \subseteq V' \cup V(E')$ does not contain $v \in V'$, then edge vv' is not covered by C . Therefore, if the vertex cover $V' \cup V(E')$ is not minimal, keep removing vertices from $V(E') \setminus V'$ until we are left with a minimal vertex cover. \square

Let $V' \cup E'$ be a nice mds and C be a minimal vertex cover such that $V' \subseteq C \subseteq V' \cup V(E')$. Let $I = V(G) \setminus C$. Note that I is an independent set and it can be partitioned into two sets I_d and I_u , where I_d is the set of vertices dominated by V' and $I_u = I \setminus I_d$. That is, $I_d = N_G(V') \cap I$, and $I_u = I \setminus I_d$. Also, let $Z = C \setminus V'$. We thus have a partition of $V(G)$ into V', Z, I_d and I_u . We call the quadruple (V', Z, I_d, I_u) a nice partition of $V(G)$ with respect to the mds $V' \cup E'$ and the minimal vertex cover C (see Fig. 1). Also, we refer to the graph $G' = G[Z \cup I_u]$ as the companion graph of G with respect to V' and C .

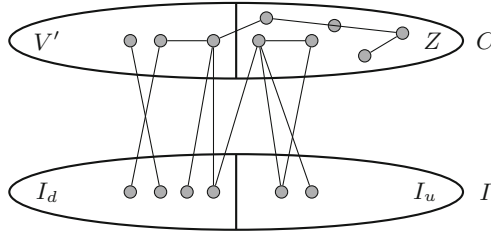


Fig. 1. Partition of $V(G)$ into minimal vertex cover C and independent set I , where $C = V' \uplus Z$ and $I = I_d \uplus I_u$.

Now let us define a new kind of domination called *special domination*. We say a vertex *special dominates* only itself, and an edge *special dominates* its endpoints as well as all the edges incident to at least one of its endpoints. Consequently, we can define a *special dominating set (sds)* as follows. A *sds* of a graph G' is a set $Q' \subseteq V(G') \cup E(G')$ such that every element $x \in (V(G') \cup E(G')) \setminus Q'$ is either adjacent to or incident on an edge in Q' . The next lemma shows the relation between mds and sds.

Lemma 3. *Let $V' \cup E'$ be a nice mds of G and C be a minimal vertex cover of G such that $V' \subseteq C \subseteq V' \cup V(E')$. Let (V', Z, I_d, I_u) be a nice partition of $V(G)$ with respect to $V' \cup E'$ and C . Then G has a mds of size at most k if and only if $G' = G[Z \cup I_u]$ has a sds of size at most $k - |V'|$.*

Proof. Assume G has a mds of size at most k . Since $V' \cup E'$ is a nice mds, $|V' \cup E'| \leq k$. We can construct a sds Q' for G' as follows. If an edge $e \in E'$ has both its endpoints in $V(G')$, add e to Q' . If an edge $e \in E'$ has exactly one endpoint in $V(G')$, then add that endpoint to Q' .

We now claim that Q' is indeed a sds for G' . Since E' dominates every vertex in $V(E') \supseteq Z \cup I_u = V(G')$, Q' special dominates all vertices of G' . If $e = uv$ is an edge of G' such that there exists an edge $uw \in E'$ (or $vw \in E'$) for some $w \in V(G')$, then $uw \in Q'$ (or $vw \in Q'$) and hence Q' special dominates e .

We claim that Q' special dominates all the edges in G' . By way of contradiction, suppose $e = uv$ is an edge of G' such that there is no edge uw' or vw' in E' for any $w' \in V(G')$. Note that this also means $uv \notin E'$. But $u, v \in V(E') \supseteq Z \cup I_u$. In that case, there must exist $xu, yv \in E'$, where

$x, y \in V' \cup I_d$. Then we claim that $S = V' \cup ((E' \setminus \{xu, yv\}) \cup \{uv\})$ is a mds of G . Notice that $\{xu, yv\}$ dominate the set of vertices $R = \{x, u, y, v\}$ and all the edges $E(R)$ incident to at least one vertex in $R = \{x, u, y, v\}$. Since $V' \cup E'$ is a mds of G , to prove S is a mds of G , it is enough to show that S dominates R and $E(R)$. Since $S \supseteq V' \cup \{uv\}$ and $x, y \in I_d$, we have that $x, y \in N_G[V']$. This implies that S dominates R . Now, what is left to prove is, S dominates $E(R)$. Since $uv \in S$, all the edges incident to at least one of u or v is dominated by S . Finally, we show that S dominates all the edges incident to at least one of x or y . Let e be an edge incident on $z \in \{x, y\}$. If $z \in V'$, then S dominates e , because $z \in V' \subseteq S$. Otherwise $z \in I_d$, because $z \in \{x, y\} \subseteq V' \cup I_d$. Let $e = zw$. Since $z \in I_d$ and $V' \cup Z$ is a vertex cover of G , we have that $w \in V' \cup Z$. If $w \in V'$, then S dominates $e = zw$, because $w \in V' \subseteq S$. If $w \in \{u, v\}$, then S dominates $e = zw$, because $uv \in S$. Otherwise $w \in Z \setminus \{u, v\} \subseteq V(E') \setminus \{u, v\}$. Since $w \in V(E') \setminus \{u, v\}$, there is an edge in $E' \setminus \{xu, yv\} \subseteq S$. This implies that S dominates e . Thus we have shown that S is a mds of cardinality strictly less than that of $V' \cup E'$, a contradiction. Hence we conclude that Q' is a sds of G' .

To prove the other direction, suppose G' has a sds Q' of size at most $k - |V'|$. We claim that $V' \cup Q'$ is an mds of G . Note that Q' dominates all vertices and edges in graph G' as well as all edges incident on $Z \cup I_u$, and V' dominates all vertices in $V' \cup I_d$ as well as all edges incident on V' . Therefore, $V' \cup Q'$ is a mds of G of cardinality $|V'| + |Q'| \leq k$. □

Lemma 3 shows that given a graph G , V' and C as defined above, the problem of deciding whether G has a mds of size at most k boils down to deciding whether G' has a sds of size at most $k - |V'|$. This results in solving the following problem.

SPECIAL DOMINATING SET (SDS)

Input: An undirected graph G and a positive integer ℓ .

Question: Does there exist a sds of size at most ℓ in G ?

In what follows we first design a polynomial time algorithm for SDS. Towards this, note that an edge has more “special dominating power” than a vertex has, in the sense that an edge special dominates itself, its endpoints and its adjacent edges, whereas a vertex special dominates *only* itself. Therefore, a natural strategy is to first try to special dominate as many vertices and all edges with as *few edges* as possible, and then add to the solution all those vertices that are not special dominated by any of the edges. This intuition leads to following polynomial time algorithm for SDS.

ALGORITHM-SDS (G, ℓ)

Step 1. Find a maximum matching, say M , in G . Let $U = V(G) \setminus V(M)$.

Step 2. If $|M \cup U| \leq \ell$, return **Yes**; else return **No**.

The only time consuming step in the above algorithm is Step 1 – finding a maximum matching – and this can be done in time $\mathcal{O}(m\sqrt{n})$ [21]. Thus, ALGORITHM-SDS runs in polynomial time, and the following lemma shows the correctness of the algorithm.

Lemma 4. *Let M be a maximum matching in a graph G and let $U = V(G) \setminus V(M)$. Then $M \cup U$ is a minimum sized sds of G .*

Proof. Since M is a maximum matching, every edge $e \in E \setminus M$ is incident to an edge in M , and thus M special dominates all edges in G . The set M also special dominates all vertices in $V(M)$, and the rest of the vertices in G are special dominated by U . Therefore, $M \cup U$ is indeed a sds of G .

Now we claim that $M \cup U$ is a minimum size sds of G . Since $V(M) \cap U = \emptyset$ and $V(G) = V(M) \cup U$, we have that $|V(G)| = 2|M| + |U|$. Towards proving the minimality of $M \cup U$, we show that any sds $E_1 \cup V_1$ of G , where $E_1 \subseteq E(G)$ and $V_1 \subseteq V(G)$, has cardinality at least $|M \cup U| = |M| + |U|$. Let M_1 be a maximum (w.r.t. E_1) matching contained in E_1 . The total number of vertices special dominated by E_1 is at most $2|M_1| + |E_1 \setminus M_1| \leq |M_1| + |E_1|$. Since $E_1 \cup V_1$ is a sds of G , we have

$$\begin{aligned} |M_1| + |E_1| + |V_1| &\geq |V(G)| \\ |E_1| + |V_1| &\geq |V(G)| - |M_1| \\ &\geq 2|M| + |U| - |M_1| && \text{(because } |V(G)| = 2|M| + |U| \text{)} \\ &\geq |M| + |U|. && \text{(because } |M| \geq |M_1| \text{)} \end{aligned}$$

This completes the proof of the lemma. □

ALGORITHM-SDS together with Lemma 4 results in the following result.

Lemma 5. *SDS can be solved in time $\mathcal{O}(m\sqrt{n})$.*

We are now fully equipped to give our algorithm for MDS.

ALGORITHM-MDS (G, k)

Step 1. Enumerate all minimal vertex covers of G of size at most $2k$. Let \mathcal{C} be the collection of such vertex covers.

Step 2. For each $C \in \mathcal{C}$ and each $V' \subseteq C$ such that $|V'| \leq k$ and $|C| \leq 2k - |V'|$, use ALGORITHM-SDS to decide if the companion graph G' (w.r.t. C and V') has a sds of size at most $k - |V'|$; if it has, return **Yes**.

Step 3. Otherwise return **No**.

The correctness of the algorithm follows from Lemma 3. Now, let us analyze the running time of ALGORITHM-MDS. Any graph has at most 2^{2k} minimal vertex covers of size at most $2k$. Furthermore, given G and k , all minimal vertex covers of size at most $2k$ can be enumerated in time $2^{2k}n^{\mathcal{O}(1)}$ [10]. This means, Step 1 can be executed in time $2^{2k}n^{\mathcal{O}(1)}$.

For each $C \in \mathcal{C}$, there are at most $2^{|C|} \leq 2^{2k}$ choices for V' . For each such choice of C and V' , by Lemma 5, a minimum sds in G' can be found in polynomial time. Therefore, the running time of ALGORITHM-MDS (G, k) can be bounded by $2^{2k} \cdot 2^{2k} \cdot n^{\mathcal{O}(1)} = \mathcal{O}^*(16^k)$. This, however, is a liberal estimate. A finer analysis shows that the running time can be brought down to $\mathcal{O}^*(7.465^k)$.

Lemma 6. ALGORITHM-MDS runs in time $\mathcal{O}^*(7.465^k)$.

Proof. If (G, k) is a yes-instance of MDS with a nice mds $V' \cup E'$, where $|V'| = j$, then any minimal vertex cover C such that $V' \subseteq C \subseteq V(E')$ can have size at most $|V'| + 2|E'| \leq j + 2(k - j) = 2k - j$. Therefore, in Step 2, we only process those pairs (C, V') such that $|C| \leq 2k - j$, where $|V'| = j$, and there are only 2^{2k-j} such C . Thus Step 2 takes time

$$\sum_{j=0}^k 2^{2k-j} \binom{2k-j}{j} = 2^{2k} \sum_{j=0}^k 2^{-j} \binom{2k-j}{j}.$$

Since for any $x > 0$, $\binom{n}{i} x^i \leq \sum_{i'=0}^n \binom{n}{i'} x^{i'} = (1+x)^n$, we get $\binom{n}{i} \leq (1+x)^n / x^i$. Using this inequality, for any $x > 0$,

$$2^{-j} \binom{2k-j}{j} \leq \frac{(1+x)^{2k-j}}{(2x)^j} = \frac{(1+x)^{2k}}{((1+x) \cdot 2x)^j}.$$

We choose $x = \frac{(\sqrt{3}-1)}{2}$ so that $(1+x) \cdot 2x = 1$. This gives $\frac{(1+x)^{2k}}{((1+x)2x)^j} \leq (1.3661)^{2k}$. Hence, Step 2 can be executed in time $2^{2k} \cdot 1.3661^{2k} \cdot n^{\mathcal{O}(1)} \leq (7.465)^k \cdot n^{\mathcal{O}(1)}$. \square

Thus, we get the following theorem.

Theorem 1. MDS parameterized by k can be solved in time $\mathcal{O}^*(7.465^k)$.

4 Outline of Algorithm for MDS Parameterized by Treewidth

In this section we only give a brief outline of our algorithm for MDS parameterized by treewidth of the input graph. Due to space constraint, we omit the complete description of the algorithm and its analysis. Here, the input is graph G and a tree decomposition of G of width $\text{tw}(G)$.

To design an algorithm we first prove that there is a minimum sized mixed dominating set S of G such that (i) the edges in S form a matching, and (ii) the set of endpoints of the edges in S is disjoint from the vertices in S .

We now give an informal description of our algorithm. Let G be the input graph and $(\mathbb{T}, \mathcal{X} = \{X_t\}_{t \in V(\mathbb{T})})$ be the given tree decomposition of G . Any standard dynamic programming over tree decomposition has three ingredients: for any node $t \in \mathbb{T}$ (i) defining partial solution, (ii) defining equivalence classes among partial solutions (or in other words defining states of DP table according to partial solutions), and (iii) computing a ‘best partial solution’ for each state from previously computed values. Normally, for any node $t \in \mathbb{T}$, partial solutions are defined according to the properties of the intersection of solutions with the graph G_t . In our case, a partial solution will be a subset of $V(G_t) \cup E(G_t)$. When we define equivalence classes of partial solutions, one of the factors under consideration is the intersection of these partial solutions with the bag X_t . Since

partial solutions contain edges, at the first blush, the number of choices for these partial solutions seems to be at least $2^{O(tw^2)}$. Instead, we prove that it has an equivalent characterization in terms of pairs of vertices which allows us to bound the number equivalence classes. Recall that there is a minimum mixed dominating set $S = V' \cup E'$, where $V' \subseteq V(G)$ and $E' \subseteq E(G)$, with the following properties:

- (a) E' is a matching, and
- (b) $V' \cap V(E') = \emptyset$.

Let $V' \cup E'$ be a solution satisfying conditions (a) and (b). Consider the pair $(V', V(E'))$. Since $V' \cup E'$ is a solution, we have that (i) $(V', V(E'))$ is a vertex cover of G , (ii) $N_G[V'] \cup V(E') = V(G)$, and (iii) $G[V(E')]$ has a perfect matching. In fact, one can show that any pair of vertex subsets that satisfies these three properties can be turned into a mixed dominating set. That is, these two notions are *equivalent*. As a result, for any node t in the tree decomposition we define partial solutions and equivalence classes among partial solutions as follows. A *partial solution* is a tuple (X, F, Y) satisfying the following conditions, where $X \subseteq V(G_t)$, $F \subseteq E(G_t)$, $Y \subseteq X_t$:

- $X \uplus Y \uplus V(F)$ is a vertex cover of G_t ,
- $V(G_t) \setminus X_t \subseteq N_{G_t}[X] \cup V(F)$.

The intuitive meaning of (X, F, Y) is that there will potentially be a solution S such that $X \cup F \subseteq S$ and for each $u \in Y$, there will be an edge $uv \in S \setminus E(G_t)$.

We now define equivalence classes of partial solutions corresponding to a node t in the tree decomposition. We define $\mathcal{P}_t[f]$, where $f : X_t \rightarrow \{1, 2, 2', 3, 3'\}$ as the set of partial solutions (X, F, Y) , which satisfy the following.

1. $X_t \cap X = f^{-1}(1)$,
2. $X_t \cap V(F) = f^{-1}(2)$,
3. $Y = f^{-1}(2')$, and
4. $(N_{G_t}(X) \cap X_t) \setminus (Y \cup V(F)) \supseteq f^{-1}(3)$.

Informally, each partial solution imposes a partition of X_t , which is defined by f . The set $f^{-1}(1)$ is the set of vertices from X_t which are part of solution. The set $f^{-1}(2)$ is the set of vertices from X_t such that there are edges in the solution which are incident on vertices in $f^{-1}(2)$ and are present in the graph G_t . The set $f^{-1}(2')$ is the set of vertices from X_t such that there are edges in the solution which are incident on vertices in $f^{-1}(2')$ and these edges are not present in the graph G_t . Here, the condition 4 implies that the set $f^{-1}(3)$ is the set of vertices in X_t , which are not part of solution vertices or end points of solution edges in the partial solution, but they are already dominated. The set $f^{-1}(3')$ is the set of vertices in X_t which are not yet dominated and not in $f^{-1}(2')$. The number of equivalence classes is bounded by $5^{tw(G)}$. Thus, a standard dynamic programming, coupled with fast computation of cover product [3], results in the the following theorem.

Theorem 2. *Given a graph G together with a tree decomposition of width tw , MDS can be solved in time $\mathcal{O}^*(6^{\text{tw}})$.*

Theorem 3. *MDS on planar graphs can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$.*

Proof. By Planar Excluded Grid Theorem [12, 24], treewidth of a planar graph that has a vertex cover of size $2k$ is smaller than $(9/2)\sqrt{4k+2}$. So graphs of treewidth larger than $(9/2)\sqrt{4k+2}$ are **No**-instances of MDS. To exploit it algorithmically, we use the algorithm of Bodlaender et al. [4] which takes as input an n -vertex graph and an integer $\ell > 0$, runs in time $2^{\mathcal{O}(\ell)}n$, and either conclude that treewidth of G is more than ℓ or gives a tree decomposition of with $5\ell + 4$. For convenience, let us call this algorithm **ALG-A**.

We run **ALG-A** on input G and $\ell = (9/2)\sqrt{4k+2}$. If it outputs that treewidth of G is greater than ℓ , then we conclude that G is a **No**-instance of MDS. Otherwise **ALG-A** will output a tree decomposition of G of width $5\ell + 4 = (45/2)\sqrt{4k+2} + 4$. Then, we apply Theorem 2. Both algorithm **ALG-A** and the algorithm mentioned in Theorem 2 run in time $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$. This completes the proof of the theorem. \square

5 Lower Bounds

In this section first we prove the following theorem.

Theorem 4. *Unless ETH fails, MDS cannot be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\ell)})$, where ℓ is either the solution size or the treewidth of the input graph.*

Proof. Lan and Chang [17] gave a polynomial time reduction from **MODIFIED VERTEX COVER** (**MVC**), an **NP**-complete problem, to MDS on split graphs. There is a reduction from an instance (G, k) of **VERTEX COVER** to an instance (G', k') of **MVC** and a reduction from an instance of (G', k') of **MVC** to an equivalent instance (G'', k'') of **MDS** in [17]. Here, the input size and the parameter of the instances change as follows. Let $|V(G)| = n$. Then, $|V(G')| = n' \leq n + 2$, $k' = k$, $|V(G'')| = |V(G') \cup E(G')|$, $k'' = (n' + k - 1)/2 \leq (n + k + 1)/2$, where G'' is a split graph with treewidth at most n' .

ETH implies that **VERTEX COVER** on a graph with n vertices and m edges can not be solved in time $2^{\mathcal{O}(n+m)}$ [16]. As a result from the above mentioned reductions, we get that, unless **ETH** fails, MDS has no $2^{\mathcal{O}(\ell)}n^{\mathcal{O}(1)}$ algorithm, where ℓ is the solution size or treewidth of the input graph. \square

Now we prove a kernel lower bound for MDS. That is, we show that unless $\text{coNP} \subseteq \text{NP/poly}$, MDS does not admit a polynomial kernel when parameterized by k . We do this by a *polynomial parameter transformation* from an appropriate parameterization of **RED BLUE DOMINATING SET** (**RBDS**).

Definition 1 ([5]). *Let P and Q be two parameterized problems. A polynomial parameter transformation (PPT, for short) from P to Q is a polynomial time algorithm, which given an instance, say (x, k) of P , produces an equivalent instance (y, k') of Q such that $k' \leq p(k)$ for some polynomial $p(\cdot)$.*

Proposition 1 ([5]). *If there is a PPT from P to Q and P has no polynomial kernel, then Q has no polynomial kernel.*

In the RBDS problem, the input is a bipartite graph G with bipartition $R \uplus B$ and a positive integer ℓ , and the question is whether there exists a set $X \subseteq R$ of size at most ℓ , which dominates the set B , i.e., $N(X) = B$. (Such a set X is called a red-blue dominating set (rbds, for short) of G). This problem when parameterized by $|R|$ is the same as SMALL UNIVERSE HITTING SET (see [8]) and thus from [8] we get the following result.

Lemma 7 ([8]). *RBDS parameterized by $|R|$ and ℓ has no polynomial kernel unless $\text{coNP} \subseteq \text{NP/poly}$.*

Theorem 5. *MDS parameterized by the solution size has no polynomial kernel, unless $\text{coNP} \subseteq \text{NP/poly}$.*

Proof. The proof is by a polynomial parameter transformation from RBDS parameterized by $|R|$ and ℓ . Given an instance $(G = (R \uplus B, E), \ell)$ of RBDS, we construct an equivalent instance $(G', |R| + \ell + 1)$ of MDS. If $B \subseteq V(G)$ contains an isolated vertex, then note that G has no rbds (of any size), so take G' to be a $|R| + \ell + 2$ -sized matching. Otherwise, if B has no isolated vertices, then proceed as follows (see Fig. 2).

1. Add all vertices and all edges of G to G' , i.e., $V(G') \supseteq V(G)$ and $E(G') \supseteq E(G)$.
2. Corresponding to every vertex $v_i \in R$, add vertices x_i and y_i , and add edges $v_i x_i$ and $x_i y_i$ in G' .
3. Add a vertex z and add edges $z y_i$, for all y_i .
4. Add $|R| + \ell + 2$ additional neighbors to z .

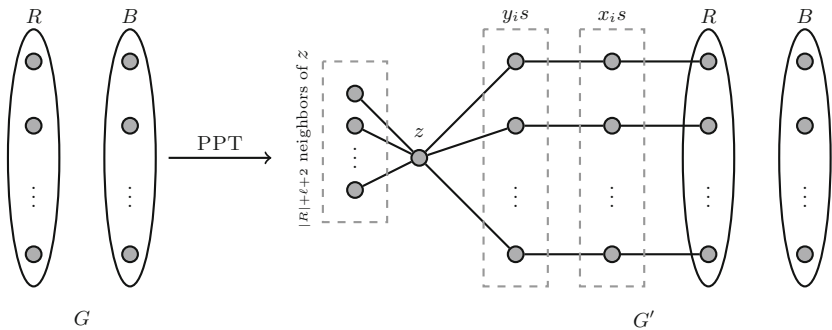


Fig. 2. PPT from RBDS to MDS

We claim that G has a rbds of size at most ℓ if and only if G' has a mds of size at most $|R| + \ell + 1$. Let $X \subseteq R$ be a rbds of G of size at most ℓ . Then $X \cup \{x_i v_i : i = 1, 2, \dots, |R|\} \cup \{z\}$ is a mds of size at most $|R| + \ell + 1$.

Conversely, assume that G does not have any rbds of size at most ℓ . Let S be a *minimum* sized mds of G' . Let S' be the set of all elements $x \in S$ such that x dominates some element(s) of B . Let $S' = S_1 \uplus S_2 \uplus S_3$, where $S_1 = S' \cap B$, $S_2 = S' \cap E(G')$ and $S_3 = S' \cap R$. Construct $S'' \subseteq R$ as follows: (i) for every $v \in S_1$, add a neighbor of v to S'' , (ii) for every edge $ww' \in S_2$, where $w \in R$ and $w' \in B$, add w to S'' , and (iii) add all vertices of S_3 to S'' . Clearly, $|S''| \leq |S'|$ and S'' is a rbds of G . By assumption, $|S''| > \ell$ which implies that $|S'| > \ell$.

Thus, S' is a subset of the minimum sized mds S and $|S'| > \ell$. Assume that $z \in S$, otherwise $|S| \geq |R| + \ell + 2$. Note that neither the elements of S' nor z can dominate any of the $|R|$ edges $x_i y_i$. And at least $|R|$ elements are required to dominate all of them. Therefore,

$$\begin{aligned} |S| &\geq |\{z\}| + |\{\text{the } |R| \text{ elements that dominate edges } x_i y_i\}| + |S'| \\ &> 1 + |R| + \ell. \end{aligned}$$

That is, G' does not have a mds of size at most $|R| + \ell + 1$. Hence, the theorem follows from the given reduction, Proposition 1 and Lemma 7. \square

Now we present an improved lower bound for MDS when parameterized by the treewidth of the input graph. We can reduce an instance of SET COVER problem (U, \mathcal{F}, ℓ) to an equivalent instance of RBDS, $(R \uplus B, E, \ell)$, where $R = \mathcal{F}$ and $B = U$. Edge set E consists of edges between $F \in R$ and $x \in B$ if and only if $x \in F$. We now apply the reduction given in the proof of Theorem 5 to an instance of RBDS, $(R \uplus B, E, |R| + \ell)$ to get an equivalent instance of MDS, $(G, |R| + \ell + 1)$. Notice that graph G has treewidth at most $1 + |B| = 1 + |U|$. The Set Cover Conjecture [7] states that SET COVER cannot be solved in $\mathcal{O}^*((2 - \epsilon)^{|U|})$ time for any $\epsilon > 0$. We thus have the following theorem.

Theorem 6. *Unless the Set Cover Conjecture fails, MDS does not admit an algorithm with running time $\mathcal{O}^*((2 - \epsilon)^{\text{tw}(G)})$.*

6 Conclusion

In this paper we initiated a systematic study of MDS from the viewpoint of parameterized complexity and designed algorithms parameterized by the solution size and the treewidth of the input graph. The algorithm for MDS parameterized by the treewidth significantly improved the known algorithm for the problem. It is curious to note that our algorithm runs in time $\mathcal{O}^*(5^{\text{pw}})$ on graphs of pathwidth pw , while the same algorithm runs in time $\mathcal{O}^*(6^{\text{tw}})$ on graphs of treewidth tw . It will be interesting to close this gap as well as prove an optimal lower bound under the Strong Exponential Time Hypothesis (SETH). Another research avenue will be to find families of graph classes on which the problem does admit polynomial kernels. Designing a non-trivial exact exponential time algorithm is another interesting problem.

References

1. Alavi, Y., Behzad, M., Lesniak-Foster, L.M., Nordhaus, E.A.: Total matchings and total coverings of graphs. *J. Graph Theor.* **1**(2), 135–140 (1977)
2. Alavi, Y., Liu, J., Wang, J., Zhang, Z.: On total covers of graphs. *Discrete Math.* **100**(1–3), 229–233 (1992)
3. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Fourier meets möbius: fast subset convolution. In: *STOC*, pp. 67–74 (2007)
4. Bodlaender, H.L., Drange, P.G., Dregi, M.S., Fomin, F.V., Lokshtanov, D., Pilipczuk, M.: A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.* **45**(2), 317–378 (2016). <http://dx.doi.org/10.1137/130947374>
5. Bodlaender, H.L., Thomassé, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.* **412**(35), 4570–4578 (2011)
6. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21275-3
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* **12**(3), 41:1–41:24 (2016)
8. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and IDS. *ACM Trans. Algorithms* **11**(2), 13:1–13:20 (2014)
9. Erdős, P., Meir, A.: On total matching numbers and total covering numbers of complementary graphs. *Discrete Math.* **19**(3), 229–233 (1977)
10. Fernau, H.: On parameterized enumeration. In: Ibarra, O.H., Zhang, L. (eds.) *COCOON 2002*. LNCS, vol. 2387, pp. 564–573. Springer, Heidelberg (2002). doi:10.1007/3-540-45655-4_60
11. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York (1979)
12. Gu, Q., Tamaki, H.: Improved bounds on the planar branchwidth with respect to the largest grid minor size. *Algorithmica* **64**(3), 416–453 (2012)
13. Hatami, P.: An approximation algorithm for the total covering problem. *Discuss. Math. Graph Theor.* **27**(3), 553–558 (2007)
14. Haynes, T.W., Hedetniemi, S., Slater, P.: *Fundamentals of Domination in Graphs*. CRC Press, Boca Raton (1998)
15. Hedetniemi, S.M., Hedetniemi, S.T., Laskar, R., McRae, A., Majumdar, A.: Domination, independence and irredundance in total graphs: a brief survey. In: *Proceedings of the 7th Quadrennial International Conference on the Theory and Applications of Graphs*. vol. 2, pp. 671–683 (1995)
16. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity. *J. Comput. Syst. Sci.* **63**(4), 512–530 (2001)
17. Lan, J.K., Chang, G.J.: On the mixed domination problem in graphs. *Theor. Comput. Sci.* **476**, 84–93 (2013)
18. Majumdar, A.: *Neighborhood hypergraphs: a framework for covering and packing parameters in graphs*. Ph.D. thesis, Clemson University (1992)
19. Manlove, D.: On the algorithmic complexity of twelve covering and independence parameters of graphs. *Discrete Appl. Math.* **91**(1–3), 155–175 (1999)
20. Meir, A.: On total covering and matching of graphs. *J. Comb. Theor. Ser. B* **24**(2), 164–168 (1978)
21. Micali, S., Vazirani, V.V.: An $\mathcal{O}(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs. In: *FOCS*, pp. 17–27 (1980)

22. Peled, U.N., Sun, F.: Total matchings and total coverings of threshold graphs. *Discrete Appl. Math.* **49**(1–3), 325–330 (1994)
23. Rajaati, M., Hooshmandasl, M.R., Dinneen, M.J., Shakiba, A.: On fixed-parameter tractability of the mixed domination problem for graphs with bounded tree-width. *CoRR* abs/1612.08234 (2016)
24. Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a planar graph. *J. Comb. Theor. Ser. B* **62**(2), 323–348 (1994)
25. Yannakakis, M., Gavril, F.: Edge dominating sets in graphs. *SIAM J. Appl. Math.* **38**(3), 364–372 (1980)
26. Zhao, Y., Kang, L., Sohn, M.Y.: The algorithmic complexity of mixed domination in graphs. *Theor. Comput. Sci.* **412**(22), 2387–2392 (2011)