RAIRO-Oper. Res. 59 (2025) 3267–3284
https://doi.org/10.1051/ro/2025125

RAIRO Operations Research
www.rairo-ro.org

ALGORITHMS FOR MINIMUM MEMBERSHIP DOMINATING SET PROBLEM

Sangam Balchandar Reddy and Anjeneya Swami Kare*

Abstract. Given a graph $G = (V, E)$ and an integer k, the Minimum Membership Dominating Set (MMDS) problem asks to compute a dominating set $S \subseteq V$ such that for each vertex $u \in V$, $1 \leq |N[u] \cap S| \leq k$. The problem is known to be NP-complete even on split graphs and planar bipartite graphs. In this paper, we approach the problem from an algorithmic standpoint and obtain several interesting results. We give an $\mathcal{O}^*(1.5719^n)$ time exact algorithm for the problem on split graphs. Following a reduction from a special case of the 1-in-3 SAT problem, we show that there is no sub-exponential time algorithm that runs in time $\mathcal{O}^*(2^{o(n)})$ for bipartite graphs, for any $k \geq 2$. We also prove that the problem is NP-complete when $\Delta = k + 2$, for any $k \geq 5$, even for bipartite graphs. We investigate the parameterized complexity of the problem for the parameter twin cover number and the combined parameter distance to cluster, membership(k) and prove that the problem is fixed-parameter tractable. Using a dynamic programming based approach, we obtain a linear-time algorithm for trees.

Mathematics Subject Classification. 05C85, 05C05, 68Q25, 68Q27, 68R10.

Received December 18, 2024. Accepted September 9, 2025.

1. Introduction

Given a graph $G = (V, E)$, a set $D \subseteq V$ is a *dominating set*, if each vertex $u \in V$ is either in D or has a neighbour in D. Similarly, a set $D \subseteq V$ is a *total dominating set*, if each vertex $u \in V$ has a neighbour in D. The Dominating Set problem is one of the 21 NP-complete problems introduced by Karp [14]. In this paper, we consider a variant of the Dominating Set problem called Minimum Membership Dominating Set (MMDS). The MMDS problem seeks to compute a dominating set $S \subseteq V$ such that for each vertex $u \in V$: $1 \leq |N[u] \cap S| \leq k$. The problem is defined as follows.

Problem. Minimum Membership Dominating Set
Input. A graph $G = (V, E)$ and a positive integer k.
Parameter. k
Question. Does there exist a *dominating set* $S \subseteq V$ such that $1 \leq |N[u] \cap S| \leq k$ for each vertex $u \in V$?

MMDS is a decision problem and the term **Minimum** in Minimum Membership Dominating Set does not indicate that it is a minimization problem. We simply reuse the problem statement as defined in [2].

Keywords. Dominating set, minimum membership dominating set, exact algorithms, bounded degree graphs, FPT, trees.

School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India.
*Corresponding author: askcs@uohyd.ac.in

Known results. Kuhn *et al.* [17] initiated the study on the membership version of the SET COVER problem called MINIMUM MEMBERSHIP SET COVER. They have proved that the problem is NP-complete and also obtained a lower bound on the approximation factor, which is $\mathcal{O}(\log n)$. Similarly, the MINIMUM MEMBERSHIP HITTING SET has been introduced by Narayanaswamy *et al.* [19]. In the geometric setting, they have shown that the problem does not admit a $2-\epsilon$ approximation algorithm for segments intersecting segments. They also provide a polynomial time algorithm for lines intersecting segments. Agrawal *et al.* [2] defined the MMDS problem and investigated its parameterized complexity. They have proved that the problem is NP-complete on planar bipartite graphs even for $k = 1$. They have also shown that the problem admits an FPT algorithm for the parameter vertex cover. For the parameter pathwidth (hence for treewidth and clique-width), they have proved that the problem is W[1]-hard. Recently, Karthika *et al.* [15] proved that the problem is FPT parameterized by distance to disjoint paths, distance to cluster and distance to co-cluster. They also showed that the problem can be solved in time $k^{5cw} \cdot n^{\mathcal{O}(1)}$, where cw is the clique width of the input graph.

The DOMINATING SET problem is known to be NP-complete on split graphs and bipartite graphs [5]. Fomin *et al.* [10] gave exact algorithms for bipartite graphs and split graphs with a running time of $\mathcal{O}^*(1.73206^n)$ and $\mathcal{O}^*(1.41422^n)$ respectively. For general graphs, their approach leads to an exact algorithm with running time $\mathcal{O}^*(1.93782^n)$. Later, Liedloff *et al.* [18] provided an algorithm that runs in time $\mathcal{O}^*(1.4143^n)$ on bipartite graphs and $\mathcal{O}^*(1.7088^n)$ on general graphs. Fomin *et al.* [9] improved the running time to $\mathcal{O}^*(1.5264^n)$ using a measure and conquer approach. A $\mathcal{O}^*(1.4969^n)$ time exact algorithm was presented by van Rooij [22]. Iwata *et al.* [13] provided the best known exact algorithm for general graphs that runs in $\mathcal{O}^*(1.4864^n)$ time and polynomial space. The result was achieved by developing a new analyzing technique called the "potential method". Recently, Abu-Khzam *et al.* [1] presented an $\mathcal{O}^*(1.3384^n)$ algorithm for chordal graphs using the concept of simplicial vertices. Kikuno *et al.* [16] proved that the DOMINATING SET problem is NP-complete on cubic planar graphs. Alber *et al.* [4], obtained an $\mathcal{O}^*(4^{tw})$ time algorithm for DOMINATING SET problem, where tw is the treewidth of the input graph. Later, van Rooij *et al.* [23] proposed an algorithm with running time $\mathcal{O}^*(3^{tw})$.

Given a graph $G = (V, E)$, $[i, j]$-DOMINATING SET problem asks to compute a set $D \subseteq V$, such that each vertex in $V \setminus D$ has at least i and at most j neighbours in D. $[i, j]$-TOTAL DOMINATING SET problem asks to compute a set $D \subseteq V$, such that each vertex in V has at least i and at most j neighbours in D. Goharshady *et al.* [12] proposed linear-time algorithms for $[1, 2]$-DOMINATING SET and $[1, 2]$-TOTAL DOMINATING SET problems on trees. Chellali *et al.* [6] have proved that $[1, 2]$-Dominating set problem is NP-complete even on bipartite graphs. They have also provided bounds for various graph classes such as grid graphs, P_4-free graphs, and caterpillars. Meybodi *et al.* [3] studied the parameterized complexity of $[1, j]$-DOMINATING SET and $[1, j]$-TOTAL DOMINATING SET problems and provided the lower bounds for split graphs (under ETH) and parameter pathwidth (under SETH). $[1, j]$-DOMINATING SET and $[1, j]$-TOTAL DOMINATING SET problems are known to be solvable in time $\mathcal{O}^*((j + 2)^{tw})$ and $\mathcal{O}^*((2j + 2)^{tw})$, respectively, on graphs width treewidth at most tw [23].

The MINIMUM MEMBERSHIP DOMINATING SET problem differs from the well studied $[1, j]$-TOTAL DOMINATING SET problem. In MMDS, we try to find whether their exists a *minimum membership dominating set* irrespective of the size, but in case of $[1, j]$-TOTAL DOMINATING SET the solution set has to be of minimum size.

Our results. As MINIMUM MEMBERSHIP DOMINATING SET is NP-complete on planar bipartite graphs even for $k = 1$, it is para-NP-hard for the membership parameter. Hence, we focus on various structural parameters and also provide exact algorithms. We obtain the following results.

(1) We provide an $\mathcal{O}^*(1.5719^n)$ time exact algorithm for split graphs.
(2) Assuming ETH, we show that no sub-exponential time algorithm exists for the problem on bipartite graphs, for $k \geq 2$.
(3) We study the problem on bounded degree graphs and show that the problem is NP-complete when $\Delta = k+2$, for any $k \geq 5$, even for bipartite graphs.
(4) We investigate the parameterized complexity of the problem and prove that the problem is FPT when parameterized by twin cover number and the combined parameter distance to cluster, membership(k).

(5) Using a dynamic programming based approach, we provide a linear-time algorithm for trees.

Notations. We consider only simple, finite, connected, and undirected graphs. Let $G = (V, E)$ be a graph with V as the vertex set and E as the edge set, such that $n = |V|$ and $m = |E|$. The set of vertices that belong to $N(u)$ and $N[u]$, respectively, are referred to as the neighbours and closed neighbours of a vertex u. The open neighbourhood of a set $T \subseteq V$ is denoted by $N(T)$ and the closed neighbourhood by $N[T]$. $N(T) = \bigcup_{u\in T} N(u)$ and $N[T] = \bigcup_{u\in T} N[u]$. The collective neighbourhood of a set T in another set D is the set of vertices in $N[T] \cap D$. Two vertices u and v are said to be true twins, if they have the same closed neighbourhood, *i.e.*, $N[u] = N[v]$. The degree of a vertex u is represented by $\deg(u)$ and $\deg(u) = |N(u)|$. A graph is cubic if each vertex has a degree of three. Throughout this paper, we use S to denote a *minimum membership dominating set*. We say that a vertex $u \in V$ satisfies the MMDS constraint, if $1 \leq |N[u] \cap S| \leq k$. Similarly, we say that a set $T \subseteq V$ satisfies the MMDS constraint if, for every vertex $u \in T$, $1 \leq |N[u] \cap S| \leq k$. In addition to this, we use the standard notation as defined in [24].

We use $\mathcal{O}^*(f(n))$ to denote the time complexity of the form $\mathcal{O}(f(n) \cdot n^{\mathcal{O}(1)})$. A problem is considered to be *fixed-parameter tractable* w.r.t. a parameter k, if there exists an algorithm with running time $\mathcal{O}(f(k)n^{\mathcal{O}(1)})$, where f is some computable function. Similarly, exact exponential algorithms run in time $\mathcal{O}^*(c^n)$ where c is some constant and $c > 1$. For more information on *parameterized complexity* and *exact algorithms*, we refer the reader to [7, 8], respectively.

For $q \geq 3$, let δ_q be the infimum of the set of constants c for which there exists an algorithm that solves q-SAT with n variables and m clauses in $2^{cn} \cdot m^{\mathcal{O}(1)}$ time. The **Exponential-time Hypothesis (ETH)** conjectures that $\delta_3 > 0$ where as the **Strong Exponential-time Hypothesis (SETH)** conjectures that $\lim_{q\to\infty} \delta_q = 1$. In other words, **ETH** conjectures that there is some $\epsilon > 0$ such that 3-SAT cannot be solved in time $\mathcal{O}(2^{\epsilon n})$ and **SETH** conjectures that for all $0 < \epsilon < 1$, there exists some $q \geq 3$ such that q-SAT cannot be solved in time $\mathcal{O}(2^{(1-\epsilon)n})$.

2. Preliminary results

In this section, assuming ETH, we show that a variant of the 3-SAT problem, that is, 3-CNF$^{\leq 3}$-XSAT does not admit an $\mathcal{O}^*(2^{o(n)})$ time algorithm. This result helps to prove some of the key results of the paper.

We first define a SAT variant, XSAT as follows.

XSAT problem: Given a CNF formula, the problem is to determine whether there exists an assignment such that each clause contains exactly one true literal.

We add a list of additional constraints on the CNF formula to obtain k-CNF, k-CNF$^{\leq l}$, k-CNF$_+$ and k-CNF$_+^{\leq l}$. The definitions of these variants of CNF are given as follows.

k-CNF: A CNF formula that contains exactly k literals in each clause.
k-CNF$^{\leq l}$: A CNF formula that contains exactly k literals in each clause and each variable occurs in at most l clauses.
k-CNF$_+$ and k-CNF$_+^{\leq l}$ have an additional constraint on k-CNF and k-CNF$^{\leq l}$, respectively, that all the literals are positive.

We define the problems k-CNF-XSAT and k-CNF$^{\leq l}$-XSAT as follows:

k-CNF-XSAT problem: Given a k-CNF formula, the problem is to determine whether there exists an assignment such that each clause contains exactly one true literal.

k-CNF$^{\leq l}$-XSAT problem: Given a k-CNF formula with each variable occurring in at most l clauses, the problem is to determine whether there exists an assignment such that each clause contains exactly one true literal.

From [21], we have that the problem k-CNF-XSAT is NP-complete for any value of $k \geq 3$. There also exists a simpler transformation from 3-SAT to 3-CNF-XSAT. Consider an instance of the 3-SAT problem with m

clauses and n variables. For a fixed value of $k = 3$, we replace the clauses of the form: $(x \vee y \vee z)$ with $(\bar{x} \vee a \vee b)$, $(y \vee b \vee c)$ and $(\bar{z} \vee c \vee d)$ clauses of 3-CNF-XSAT. Here, we use at most three clauses and at most four fresh variables for each clause of 3-SAT. We will have a total of at most $3m$ clauses and at most $n + 4m$ variables in the reduced instance of 3-CNF-XSAT. As the reduction preserves the linearity of the size, we conclude that 3-CNF-XSAT also satisfies ETH.

Theorem 2.1 ([20], Lem. 4)**.** *k-CNF$_+^{\leq l}$-XSAT and k-CNF$^{\leq l}$-XSAT remains NP-complete, for k, $l \geq 3$.*

Theorem 2.1 is proved by providing a polynomial time reduction from k-CNF-XSAT to k-CNF$^{\leq l}$-XSAT. Consider an instance of k-CNF-XSAT with m clauses, n variables and a total of at most mk literals. According to Theorem 2.1, the reduced instance of k-CNF$^{\leq l}$-XSAT has at most $m + 2mk$ clauses and $n + mk^2$ variables. For $k = 3$, that becomes $7m$ clauses and $n + 9m$ variables.

Theorem 2.2. *3-CNF$^{\leq 3}$-XSAT problem can not be solved in $\mathcal{O}^*(2^{o(n)})$ time.*

Proof. From Theorem 2.1, we have that 3-CNF$^{\leq 3}$-XSAT problem is NP-complete. There exists a reduction from 3-SAT to 3-CNF-XSAT with $3m$ clauses and $n + 4m$ variables. The size of the reduced instance of the reduction from 3-CNF-XSAT to 3-CNF$^{\leq 3}$-XSAT is $7m$ clauses and $n + 9m$ variables. The size of the reduced instance of 3-CNF$^{\leq 3}$-XSAT in terms of the size of 3-SAT is $21m$ clauses and $n + 31m$ variables. As the number of clauses and variables in the reduction are linear in the size of the original instance of 3-SAT; 3-CNF$^{\leq 3}$-XSAT also satisfies ETH. This concludes that, 3-CNF$^{\leq 3}$-XSAT problem can not be solved in $\mathcal{O}^*(2^{o(n)})$ time. □

3. Exact algorithm for split graphs

A graph G is a split graph if its vertices can be partitioned into $V_1 \uplus V_2$, such that $G[V_1]$ is a complete graph and V_2 is an independent set in G. In this section, we present an $\mathcal{O}^*(1.5719^n)$ time algorithm for Minimum Membership Dominating Set on split graphs. We achieve this by posing a section of the problem as an instance of the Set Cover problem.

Set Cover problem

Given a universe $\mathcal{U}$ and a family of sets $\mathcal{F}$, the Set Cover problem computes a minimum number of sets from $\mathcal{F}$ that covers $\mathcal{U}$. The decision version of the Set Cover problem is defined as follows.

Input: A universe $\mathcal{U}$, a family $\mathcal{F}$ over $\mathcal{U}$, and an integer k.
Question: Does there exist a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k such that $\bigcup \mathcal{F}' = \mathcal{U}$?

Theorem 3.1 ([8], Thm. 6.10)**.** *There exists an $\mathcal{O}(1.2353^{|\mathcal{U}|+|\mathcal{F}|})$ time exact algorithm to solve* Minimum Set Cover *problem.*

Consider a split graph G. Let I denote the independent set and C denote the clique of G. If $|C| \leq k$, then the union of the vertices of C and all the vertices of I that have no neighbour in C forms a *minimum membership dominating set*. In this case, it will be an YES-instance. Hence, we assume that $|C| \geq k + 1$. We define two sets S_C and S_I, where $S_C = S \cap C$ and $S_I = S \cap I$. Although the size of C is greater than k, $|S_C|$ will be at most k (according to the MMDS constraint). We have the following two cases based on the size of I: (1) $|I| \leq 0.3476n$ and (2) $|I| > 0.3476n$.

Case 1. $|I| \leq 0.3476n$ and $|C| > 0.6524n$.

We guess the vertex set S_I of size ranging from 0 to $0.3476n$. This can be done in $\mathcal{O}(2^{0.3476n})$ ways. We find a vertex from C that has the highest number of neighbors in S_I. Let this vertex be p and let the number of neighbours of p in S_I be x. If $x > k$, we simply reject the guess. If $x = k$, we conclude that $S_C = \phi$ and terminate the process. If $x \leq k - 1$, we continue the approach in the following way. The vertex p is adjacent to x vertices from S_I and in order to satisfy the MMDS constraint for p, $|S_C|$ must be at most $k - x$. We remove

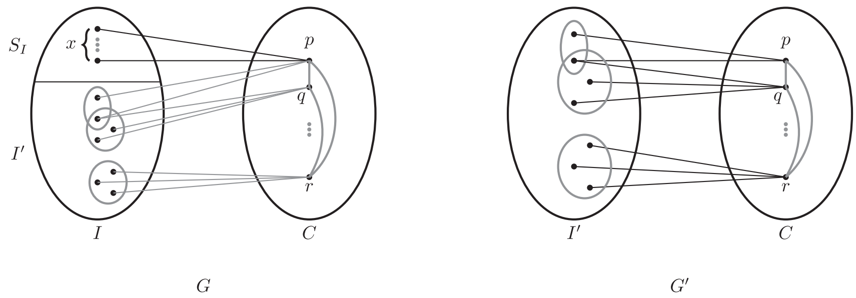

FIGURE 1. Posing a section of the MMDS problem as an instance of the SET COVER.

the vertex set that is guessed to be S_I. We have a new instance G' in which the independent set I' has only undominated vertices and some subset of vertices of C are dominated. In the end, we only have at most $k-x$ vertices in S_C, and no vertices from G' will violate the MMDS constraint.

We pose this problem as a SET COVER instance as follows. See Figure 1 for an illustration.

Input: A universe $\mathcal{U} = I'$ and sets $\mathcal{F} = \{N(v) \cap I' : v \in C\}$.
Question: Is there an $\mathcal{F}' \subseteq \mathcal{F}$ of size at most $k-x$, such that $\bigcup \mathcal{F}' = \mathcal{U}$?

Each set in the SET COVER instance corresponds to a unique vertex in C and the size of the universe is I', which is at most the size of I. Therefore, we have $|\mathcal{F}| + |\mathcal{U}| \leq n$. From Theorem 3.1, we check whether there exists a *set cover* of size $k-x$ in $\mathcal{O}^*(1.2353^n)$ time. The vertices of C that correspond to the sets $\mathcal{F}'$ will form S_C.

Case 2. $|I| > 0.3476n$ and $k+1 \leq |C| \leq 0.6524n$.

We guess the vertex set S_C of size ranging from 0 to k. This can be done in $\mathcal{O}(2^{0.6524n})$ ways. Once S_C is fixed, we compute S_I. Let $|S_C| \geq 1$. At this point, all the vertices of C are dominated. The vertices from I with no neighbour in S_C are not yet dominated. All these vertices must dominate themselves. Hence, they form the set S_I. Let $|S_C| = 0$. As all the vertices of I are undominated, they must be a part of S. Therefore, S_I contains all the vertices of I.

Once S_I and S_C are fixed, we check whether $S_I \cup S_C$ forms a *minimum membership dominating set*, in polynomial time. Case 1 takes $\mathcal{O}^*(2^{0.3476n} \cdot 1.2353^n)$ time and Case 2 takes $\mathcal{O}^*(2^{0.6524n})$ time. The overall time complexity is $\mathcal{O}^*(2^{0.3476n} \cdot 1.2353^n)$ which is equal to $\mathcal{O}^*(2^{0.6524n})$, that is $\mathcal{O}^*(1.5719^n)$.

To summarize, we have the following theorem.

Theorem 3.2. *The* MINIMUM MEMBERSHIP DOMINATING SET *problem on split graphs with n vertices can be solved in $\mathcal{O}^*(1.5719^n)$ time.*

4. COMPLEXITY ON BIPARTITE GRAPHS

In this section, we study the complexity of MINIMUM MEMBERSHIP DOMINATING SET on bipartite graphs and obtain the following results.

(1) Assuming ETH, there exists no algorithm that runs in time $\mathcal{O}^*(2^{o(n)})$ for bipartite graphs.
(2) The problem is NP-complete for $\Delta(G) \geq k+2$, when $k \geq 5$, even for bipartite graphs.

Claim 4.1. There always exists a *minimum membership dominating set* for graphs with maximum degree, $\Delta(G) \leq k$.

Proof. We consider the following two cases based on the value of $\Delta(G)$.

$\Delta(G) \leq k-1$: The vertex set V forms a *minimum membership dominating set*. As each vertex $u \in V$ has at most k vertices in its closed neighbourhood, the MMDS constraint is satisfied for each vertex $u \in V$. Therefore, it is always an YES-instance for this case.

$\Delta(G) = k$: We include all the vertices of degree at most $k-1$ to S. As all these vertices have at most k vertices in their closed neighbourhood, they satisfy the MMDS constraint. We then find all the undominated vertices of G, let this set be denoted by P. We arbitrarily pick a vertex (say u) from P and include it in S. We then update P based on the recent addition of u to S. We perform this until P is empty. We add u to S only because it is undominated, so adding u to S will not violate MMDS constraint for any vertex of G. Hence, it is an YES-instance. □

As the problem is linear-time solvable for maximum degree, $\Delta(G) \leq k$, we study the problem complexity for higher values of $\Delta(G)$.

From Theorem 2.2, we have that 3-CNF$^{\leq 3}$-XSAT problem can not be solved in $\mathcal{O}^*(2^{o(n)})$ time. We provide a reduction from 3-CNF$^{\leq 3}$-XSAT problem to show that Minimum Membership Dominating Set problem on bipartite graphs can not be solved in $\mathcal{O}^*(2^{o(n)})$ time. The reduction also proves that the MMDS problem on graphs with $\Delta(G) \geq k+2$ is NP-complete.

Reduction

Let ϕ be an instance of 3-CNF$^{\leq 3}$-XSAT problem with n variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$ and each variable occurring in at most three clauses. We construct an instance of MMDS, $I = (G, k)$ for any $k \geq 2$ as follows.

- For each clause C_j, we create three vertices u_j, y_j and w_j. For each literal x_i, we create a vertex v_i and for each literal $\bar{x}_i$, we create a vertex $\bar{v}_i$.
- If x_i is part of a clause C_j, then v_i is made adjacent to u_j and w_j. Similarly, If $\bar{x}_i$ is part of a clause C_j, then $\bar{v}_i$ is made adjacent to u_j and w_j.
- We make u_j adjacent to y_j.
- For each vertex w_j, we create a vertex set $h_j^1, h_j^2, \ldots, h_j^{k-1}$ and we make w_j adjacent to $h_j^1, h_j^2, \ldots, h_j^{k-1}$.
- Each vertex of $h_j^1, h_j^2, \ldots, h_j^{k-1}$ has $k+1$ pendant vertices adjacent to it.
- For each vertex y_j, we create a vertex set $z_j^1, z_j^2, \ldots, z_j^k$ and we make y_j also adjacent to $z_j^1, z_j^2, \ldots, z_j^k$.
- Each vertex among $z_j^1, z_j^2, \ldots, z_j^k$ has $k+1$ pendant vertices adjacent to it.
- For each vertex v_i (or $\bar{v}_i$), we create a vertex p_i (or $\bar{p}_i$) and make both the vertices adjacent.

Variable gadget. For each vertex pair (v_i, $\bar{v}_i$), we create the vertices b_i, c_i and d_i. We also create the sets $a_i^1, a_i^2, \ldots, a_i^k$ and $f_i^1, f_i^2, \ldots, f_i^{k-1}$. Vertices v_i and $\bar{v}_i$ are made adjacent to c_i and d_i. The vertex d_i is adjacent to $f_i^1, f_i^2, \ldots, f_i^{k-1}$. The vertex c_i is adjacent to b_i. The vertex b_i is also adjacent to $a_i^1, a_i^2, \ldots, a_i^k$. Each of $f_i^1, f_i^2, \ldots, f_i^{k-1}$ and $a_i^1, a_i^2, \ldots, a_i^k$ has $k+1$ pendant vertices connected to them. See Figure 2a for an illustration.

Note: The variable gadget is created only if both the vertices v_i and $\bar{v}_i$ exist. If not, we simply avoid the creation of variable gadget for such vertex v_i or $\bar{v}_i$.

Literal gadget. For each literal v_i, we create the vertices q_i and r_i. We also create a set $s_i^1, s_i^2, \ldots, s_i^k$. The vertex p_i is adjacent to vertex q_i which is adjacent to vertex r_i. The vertex r_i also has the vertices $s_i^1, s_i^2, \ldots, s_i^k$ adjacent to it. Each of $s_i^1, s_i^2, \ldots, s_i^k$ has $k+1$ pendant vertices connected to them. See Figure 2b for an illustration.

This concludes the construction of the reduced instance I. It is to be noted that, we obtain the reduced instance G, only after having the variable gadget and literal gadget at the respective places as highlighted in Figure 3.

Lemma 4.2. *If ϕ has a satisfying assignment, then I has a minimum membership dominating set with membership value k.*

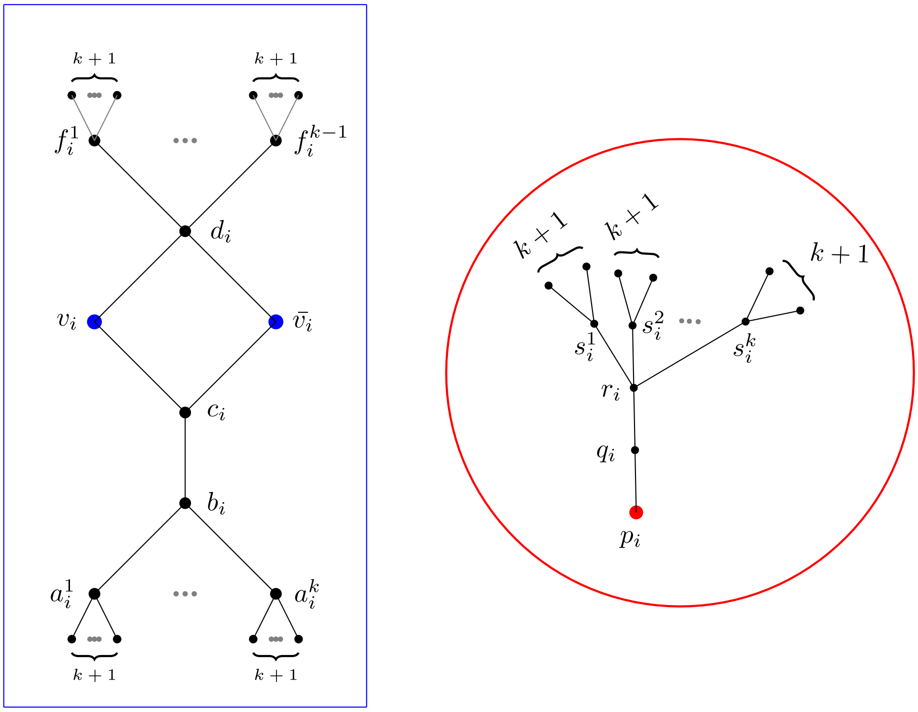

FIGURE 2. (a) Variable gadget for some $i \in [n]$ (on the *left*). (b) Literal gadget for some $i \in [n]$ (on the *right*).

Proof. Let $X = \{x_i | i \in [n]\} \to \{0, 1\}$ be a satisfying assignment for ϕ. We obtain the corresponding *minimum membership dominating set* as follows.

(1) For each $i \in [n]$:
   - Add v_i to S if $x_i = 1$ in X; add $\bar{v}_i$ to S if $x_i = 0$ in X.
   - Add $s_i^1, s_i^2, \ldots, s_i^k$ and p_i from literal gadget to S.
   - Add $f_i^1, f_i^2, \ldots, f_i^{k-1}$ and $a_i^1, a_i^2, \ldots, a_i^k$ from variable gadget to S.

(2) Add the vertices $\bigcup_{j=1}^{j=m} h_j^1 \cup h_j^2 \cup \ldots \cup h_j^{k-1}$ to S.

(3) Add the vertices $\bigcup_{j=1}^{j=m} z_j^1 \cup z_j^2 \cup \ldots \cup z_j^k$ to S.

Consider the literal gadget.

$p \in p_i$ (or $\bar{p}_i$): The vertex p has one closed neighbour in S: the vertex itself. If v_i is in S, then p has two closed neighbours in S. Therefore, p has at least one and at most two closed neighbours in S.

$p \in s_i^1, s_i^2, \ldots, s_i^k$: The vertex p has exactly one closed neighbour in S: the vertex itself and no other neighbour of p is in S.

$p \in r_i$: The k neighbours of p: $s_i^1, s_i^2, \ldots, s_i^k$ are in S. The vertex k has exactly k closed neighbours in S.

$p \in q_i$: The vertex p has exactly one neighbour, p_i in S. The vertex p has exactly one closed neighbour in S.

Consider the variable gadget.

$p \in \{f_i^1, f_i^2, \ldots, f_i^{k-1}\}$: The vertex p is in S and no other neighbours of p are in S. Hence, p has exactly one closed neighbour in S.

$p \in d_i$: The vertex p has k neighbours $f_i^1, f_i^2, \ldots, f_i^{k-1}$ in S and one among v_i and $\bar{v}_i$ in S. Therefore, p has exactly k closed neighbours in S.

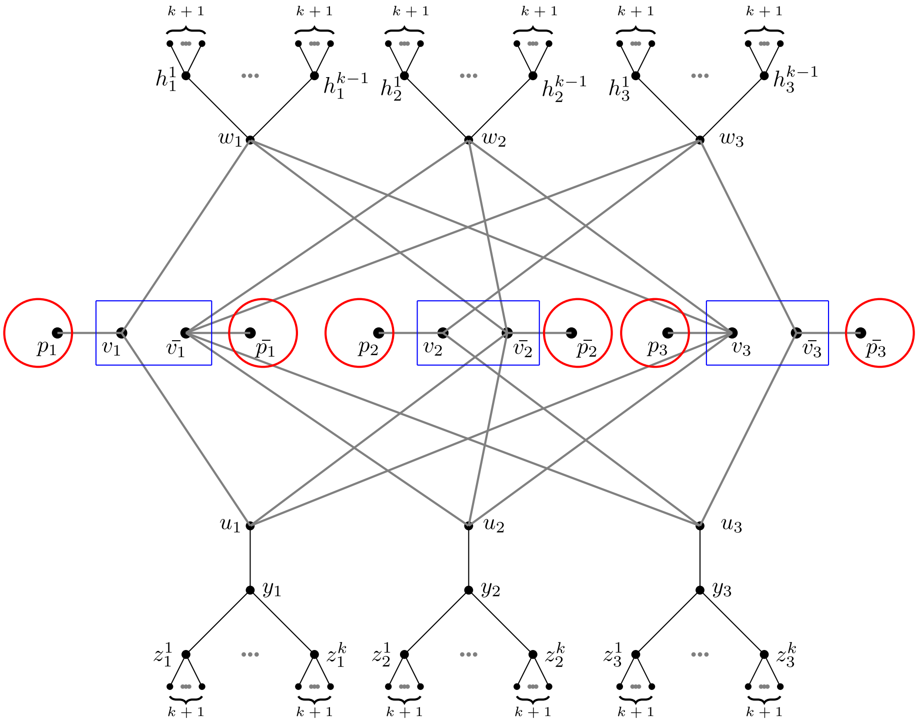

FIGURE 3. Construction of an MMDS instance from the 3-CNF$^{\leq 3}$-XSAT formula: $\phi = (x_1 \vee \bar{x_2} \vee x_3) \wedge (\bar{x_1} \vee \bar{x_2} \vee x_3) \wedge (\bar{x_1} \vee x_2 \vee \bar{x_3})$. Here, the red circles and blue rectangles are placeholders for literal gadget and variable gadget, respectively. The literal gadget and variable gadget are illustrated in Figure 2.

$p \in c_i$: One neighbour of p, either v_i or $\bar{v_i}$ is in S. As no other neighbours of p are in S, p has exactly one closed neighbour in S.

$p \in a_i^1, a_i^2, \ldots, a_i^k$: The vertex p is self dominated and no other neighbours of p are in S. Therefore, p has exactly one closed neighbour in S.

$p \in b_i$: p has exactly k neighbours, $\{a_i^1, a_i^2, \ldots, a_i^k\}$ in S.

$p \in z_j^1, z_j^2, \ldots, z_j^k$: The vertex p is self dominated and no other neighbour of p is in S. Therefore, p has exactly one closed neighbour in S.

$p \in y_j$: The vertex p has exactly k closed neighbours, $\{z_j^1, z_j^2, \ldots, z_j^k\}$ in S.

$p \in u_j$: p has exactly one neighbour from its clause variables, the one which is set to true in S and no other neighbours of p are in S. Hence, p has exactly one closed neighbour in S.

$p \in v_i$ (or) $\bar{v_i}$: The vertex p has its neighbour from the literal gadget, p_i in S. If p is a part of S, then it will have two closed neighbours in S. Otherwise, it will have exactly one closed neighbour in S.

$p \in w_j$: The vertex p has $k-1$ closed neighbours, $\{h_j^1, h_j^2, \ldots, h_j^{k-1}\}$ in S. It also has one neighbour from its clause variables in S. This leads to p having exactly k closed neighbours in S.

$p \in h_j^1, h_j^2, \ldots, h_j^{k-1}$: The vertex p is self dominated and no other neighbours of p are in S. Therefore, p has exactly one closed neighbour in S.

p is a pendant vertex: The vertex p has exactly one vertex, its neighbour, in S. Hence, p has exactly one closed neighbour in S.

As all the vertices of S satisfy the MMDS constraint, we conclude that I has a *minimum membership dominating set* with membership value k. $\square$

Lemma 4.3. *If I has a minimum membership dominating set with membership value k then ϕ has a satisfying assignment.*

Proof.

- The vertices with $k+1$ pendant vertices as neighbours must be a part of S. Therefore, the vertices $\{s_i^1, s_i^2, \ldots, s_i^k\}$; $\{f_i^1, f_i^2, \ldots, f_i^{k-1}\}$; $\{a_i^1, a_i^2, \ldots, a_i^k\}$; $\{z_j^1, z_j^2, \ldots, z_j^k\}$ and $\{h_j^1, h_j^2, \ldots, h_j^{k-1}\}$ are in S.
- The $k-1$ neighbours $\{f_i^1, f_i^2, \ldots, f_i^{k-1}\}$ of d_i are in S. Therefore, at most one among v_i and $\bar{v}_i$ can be S.
- The k neighbours $\{a_i^1, a_i^2, \ldots, a_i^k\}$ of b_i are in S, vertices b_i and c_i can not be a part of S. In order to dominate c_i, at least one among v_i and $\bar{v}_i$ can be S. This leads to having exactly one vertex from v_i and $\bar{v}_i$ in S.
- As the k neighbours $\{s_i^1, s_i^2, \ldots, s_i^k\}$ of r_i are in S, vertices r_i and q_i can not be a part of S. In order to dominate q_i; p_i must be in S.
- With the vertices $\bigcup_{i=1}^{i=n} p_i$ (or $\bar{p}_i$) being a part of S, all the vertices $\bigcup_{i=1}^{i=n}(v_i \cup \bar{v}_i)$ are dominated.
- As the $k-1$ neighbours $\{h_j^1, h_j^2, \ldots, h_j^{k-1}\}$ of w_j are in S, at most one among the three vertices of clause C_j are in S.
- Each vertex y_j has k neighbours, $\{z_j^1, z_j^2, \ldots, z_j^k\}$ in S. Hence, y_j and u_j cannot be a part of S.
- In order to dominate the vertex u_j, at least one among the three vertices of clause C_j are in S. This leads to having exactly one vertex corresponding to the clause C_j in S.

The literals corresponding to these vertices forms a satisfying assignment. Hence, we conclude that ϕ has a satisfying assignment. □

G is a bipartite graph with bipartition,

$$\left(\bigcup_{i=1}^{n} (p_i \cup \bar{p}_i \cup d_i \cup c_i \cup A_i \cup P_{F_i} \cup r_i \cup P_{S_i}) \cup P_H \cup W \cup U \cup Z\right)$$

and

$$\left(\bigcup_{i=1}^{n} (v_i \cup \bar{v}_i \cup q_i \cup P_{A_i} \cup F_i \cup b_i \cup S_i) \cup Y \cup H \cup P_Z\right),$$

where $A_i = \bigcup_{l=1}^{l=k} a_i^l$, $F_i = \bigcup_{l=1}^{l=k-1} f_i^l$, $S_i = \bigcup_{l=1}^{l=k} s_i^l$, $U = \bigcup_{j=1}^{j=m} u_j$, $Y = \bigcup_{j=1}^{j=m} y_j$, $H = \bigcup_{j=1}^{j=m} \bigcup_{l=1}^{l=k-1} h_j^l$, $Z = \bigcup_{j=1}^{j=m} \bigcup_{l=1}^{l=k} z_j^l$, $W = \bigcup_{j=1}^{j=m} w_j$; $P_{A_i}, P_{F_i}, P_{S_i}, P_H$ and P_Z represents the pendant vertices of the sets A_i, F_i, S_i, H and Z respectively.

For each value of i, one vertex pair among (p_i, v_i) and $(\bar{p}_i, \bar{v}_i)$ is in S. Irrespective of the value of k, either p_i or $\bar{p}_i$ has exactly two closed neighbours in S. Hence, $k \geq 2$.

Claim 4.4. The size of the reduced instance of I is linear in the size of the original instance ϕ.

Proof. The size of the variable gadget shown in Figure 2a is $\mathcal{O}(k^2)$. There will be at most n of them. The size of the literal gadget shown in Figure 2b is $\mathcal{O}(k^2)$. There will be at most $2n$ of them. The size of I without both the gadgets, shown in Figure 3 is $\mathcal{O}(m \cdot k^2)$. The size of the reduced instance I is $|V| = \mathcal{O}(k^2(m+n))$. If $k = \mathcal{O}(1)$, $|V| = \mathcal{O}(m+n)$. □

Hence, we have the following lower bound for the MMDS problem on bipartite graphs.

Theorem 4.5. *There exists no algorithm that runs in time $\mathcal{O}^*(2^{o(n)})$ for the* Minimum Membership Dominating Set *problem on bipartite graphs, for $k \geq 2$ (assuming that ETH holds).*

Claim 4.6. The lower bound on the maximum degree of the reduced instance is seven.

Proof. All the vertices of G, except v_i and $\bar{v_i}$, have a maximum degree of $k+2$.

A variable x_i has the following two ways of appearing in ϕ.

Case 1. Three times either in positive or negative form.

Let us assume that a variable x_i occurs three times in only positive form. This means that $\bar{x_i}$ does not exist in the formula and the variable gadget is not constructed. Vertex v_i is adjacent to three neighbours in each of $\bigcup_{j=1}^{m} u_j$ and $\bigcup_{j=1}^{m} w_j$. It is also adjacent to p_i. So, the vertex v_i has a degree of seven.

Case 2. One time in one form and two times in other form.

Let x_i occurs two times and $\bar{x_i}$ occurs one time. Vertex v_i is adjacent to two neighbours in each of $\bigcup_{j=1}^{m} u_j$ and $\bigcup_{j=1}^{m} w_j$. It is also adjacent to p_i. It also has two neighbours, c_i and d_i from the variable gadget. Vertex $\bar{v_i}$ is adjacent to one neighbour in each of $\bigcup_{j=1}^{m} u_j$ and $\bigcup_{j=1}^{m} w_j$. It is also adjacent to p_i. It also has two neighbours, c_i and d_i from the variable gadget. This leads to degree of seven and five for the vertices v_i and $\bar{v_i}$, respectively.

In both the cases, one among v_i and $\bar{v_i}$ has a degree of seven, while the other vertex has a degree less than seven. Hence, we conclude that the lower bound on the maximum degree of G is seven. □

Irrespective of the value of k, one among the two vertices p_i and $\bar{p_i}$ has a degree of seven. Therefore, the maximum degree of the reduced instance I is $k+2$ and it can not be less than seven. Hence, we arrive at the following theorem.

Theorem 4.7. *For $k \geq 5$, the* Minimum Membership Dominating Set *problem with membership k is NP-complete on bipartite graphs with maximum degree $k+2$.*

5. Parameterized complexity for structural parameters

In this section, we study the parameterized complexity of Minimum Membership Dominating Set parameterized by (1) the twin cover number and (2) the combined parameter distance to cluster, membership(k). We prove that the problem is in FPT for both the cases.

5.1. FPT parameterized by twin cover number

We consider the structural parameter twin cover number. With the help of some crucial observations, we show that the problem is in FPT. The parameter twin cover number is defined as follows.

Definition 5.1. For a graph $G = (V, E)$, an edge $(u, v) \in E$ is a twin edge of G if $N_G[u] = N_G[v]$. The parameter *twin cover number* is the cardinality of the smallest set $T \subseteq V$ such that every edge in G is either a twin edge or incident to a vertex in T.

The parameter twin cover of size at most k (if exists), can be obtained in FPT time, in the following way.

Theorem 5.2 ([11], Thm. 4)**.** *If a minimum twin cover in G has size at most k, then it is possible to compute a twin cover of size at most k in time $\mathcal{O}(|E| + k|V| + 1.2738^k)$.*

Given a graph G, we partition the vertex set V into sets T and C, where T represents the twin cover and C is the union of clique sets outside T. A clique set is a union of cliques with the same adjacency in T. We guess the set $P = T \cap S$, obtain $Q = N(P) \cap T$ and $R = T \setminus \{P \cup Q\}$. The guess of P can be made in $2^{|T|}$ ways. We check whether any vertex has more than k closed neighbours in P; if yes, we discard the guess. Let C_1 denote the union of clique sets with at least one neighbour in P and C_2 denote the union of clique sets with no neighbor in P. See Figure 4 for an illustration.

The vertices in C_2 must dominate themselves, so one vertex from each clique for all such clique sets must belong to S. We select one vertex arbitrarily to be a part of S from each clique for all these clique sets. As all the vertices of the clique have the same neighbourhood, picking any one vertex would work. Let these vertices from

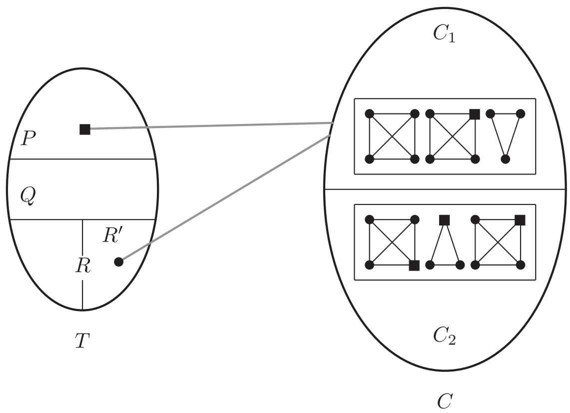

FIGURE 4. Partitioning of the vertex set V into sets T and C, where T is the twin cover and C is the union of clique sets outside T. Square shaped vertices are a part of S.

C_2 be represented by X. At this point, we again verify whether any vertex has more than k closed neighbors in S.

Currently, all the vertices of P, Q, and C are dominated and a subset of vertices of R are dominated. Obtain the vertices from R that are not yet dominated. Let these set of vertices be denoted by R'. Note that the vertices of R' do not have any neighbor in S. The only way to dominate these vertices is by having their neighbours from C_1 in S. If we choose to add vertices from a clique set $\mathbb{C} \in C_1$ to S, then adding exactly one vertex from $\mathbb{C}$ would be enough to meet our needs. We guess the clique sets from C_1, that has a vertex in S. For each clique set, there are two ways of guessing based on whether the clique set has a vertex in S. The entire guess for all the clique sets of C_1 can be made in $2^{|C_1|}$ ways, which is at most $2^{2^{|T|}}$. Let Y denote the set of vertices of this guess from C_1. We verify whether $P \cup X \cup Y$ forms a *minimum membership dominating set*, in polynomial time. The problem can be solved in $O^*(2^{|T|} \cdot 2^{2^{|T|}})$ time, where T is the twin cover size. This sums up the proof of the following theorem.

Theorem 5.3. *Given a graph $G = (V, E)$, $T \subseteq V$ is a twin cover of G, the* MINIMUM MEMBERSHIP DOMINATING SET *problem can be solved in $\mathcal{O}^*(2^{|T|} \cdot 2^{2^{|T|}})$ time.*

5.2. FPT parameterized by the combined parameter distance to cluster, membership(k)

We consider the combined parameter distance to cluster, membership(k). We propose an exponential kernel, which also infers that the problem is in FPT. The parameter distance to cluster is defined as follows.

Definition 5.4. For a graph $G = (V, E)$, the parameter *distance to cluster* is the cardinality of the smallest set $D \subseteq V$ such that every component in $V \setminus D$ is a clique.

Consider a graph G, partitioned into D and C with $|D|$ as the distance to cluster and C as the union of clique sets outside D. The collective neighbourhood of a set X in another set Y is the set of vertices in $N[X] \cap Y$. In this case, clique set represents a union of cliques such that each clique in a clique set has the same collective neighbourhood in D. We guess the subset of vertices $P = S \cap D$ and compute $Q = S \cap C$. We have $2^{|D|}$ unique subset of vertices from D. We use C_i to denote the union of clique sets with i vertices in the collective neighbourhood in D. See Figure 5 for an illustration.

For the rest of the section, we use $\mathbb{C}$ to denote an arbitrary clique set from C_i.

Consider a clique C' from $\mathbb{C}$ in which vertices u and v have the same adjacency in D.

Reduction Rule 1. If u and v are true twins, then delete either u or v.

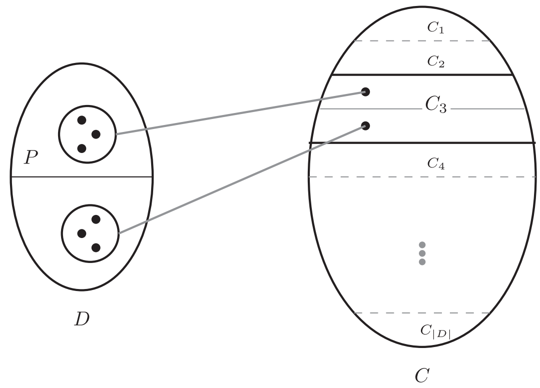

FIGURE 5. Partitioning of the vertex set V into sets D and C, where $|D|$ is the distance to cluster and C is the union of clique sets outside D. The collective neighbourhood of C_3 in D is shown.

Lemma 5.5. *Reduction Rule 1 is safe.*

Proof. As u and v are true twins, having one of them in the solution instead of both, does not alter the outcome of the MMDS constraint for any vertex in the graph. So, we arbitrarily delete one of the two vertices from the clique. □

Lemma 5.6. *The size of the largest clique in $\mathbb{C}$ is at most 2^i.*

Proof. We repeatedly apply the Reduction Rule 1 on each pair of true twins of C', and we finally end up with at most 2^i vertices that have a unique neighbourhood in D. □

The graph obtained from G, after repeatedly applying Reduction Rule 1, is denoted by G'.

We classify the cliques of C into two types. Type A: cliques that have at least one vertex not adjacent to any vertex in P. Type B: cliques that have all the vertices adjacent to a vertex in P.

First, we consider the cliques from $\mathbb{C}$ of Type A and bound them.

Lemma 5.7. *There are at most $i \cdot k$ cliques in $\mathbb{C}$ of Type A.*

Proof. Consider a clique C' from $\mathbb{C}$ that has at least one vertex (say v) not adjacent to any vertex in P. v has to be dominated by some vertex of C'. So, S consists of at least one vertex from each of these cliques. There are i vertices in D that correspond to the clique set. For all these i vertices to satisfy the MMDS constraint, each of these vertices can only be adjacent to at most k such cliques. We can have a total of at most $i \cdot k$ cliques of such sort. If there are more than $i \cdot k$ cliques, we simply reject the guess. □

Now, we consider the cliques from $\mathbb{C}$ of Type B and bound them.

Reduction Rule 2. If there are more than 2^i cliques in $\mathbb{C}$ of Type B with the same adjacency in D then delete the cliques until only 2^i cliques in $\mathbb{C}$ of Type B with the same adjacency in D remains.

Lemma 5.8. *Reduction Rule 2 is safe.*

Proof. Consider a set of cliques from $\mathbb{C}$ of Type B with the same adjacency in D. In the worst-case scenario, we might need all the vertices of a clique to dominate the vertices in D. We might end up picking 2^i vertices from 2^i unique cliques (one unique corresponding vertex from each clique). Having any more than 2^i cliques with the same adjacency in D does not alter the solution. Hence, they can be simply discarded and we end up having at most 2^i cliques of Type B with the same adjacency in D. □

After applying the Reduction Rule 2 on G', we obtain the following lemma.

Lemma 5.9. *There are at most 2^i cliques in $\mathbb{C}$ of Type B with the same adjacency in D.*

The graph obtained from G', after applying Reduction Rule 2, is denoted by G''.

Reduction Rule 3. If there are more than 2^{i2^i} cliques in $\mathbb{C}$ of Type B with a unique neighbourhood in D then delete the cliques until only 2^{i2^i} cliques in $\mathbb{C}$ of Type B with a unique neighbourhood in D remains.

Lemma 5.10. *Reduction Rule 3 is safe.*

Proof. Consider a set D' from D that corresponds to $\mathbb{C}$. The size of a clique in $\mathbb{C}$ is bounded by 2^i. We consider the presence of all possible cliques in $\mathbb{C}$ with a unique neighbourhood in D'. So, for each vertex in D', we have at most 2^{2^i} ways of forming the adjacency with the clique. This is possible for all the i vertices. This leads to a maximum of 2^{i2^i} ways of forming the adjacency between the vertices of D' and cliques in $\mathbb{C}$. This gives us 2^{i2^i} unique cliques. The total number of cliques from $\mathbb{C}$ of Type B with a unique neighbourhood in D is 2^{i2^i}. Having any more than 2^{i2^i} cliques with a unique neighbourhood in D does not alter the solution. Hence, they can be simply discarded and we end up having at most 2^{i2^i} cliques of Type B with a unique neighbourhood in D. $\square$

After applying the Reduction Rule 3 on G'', we obtain the following lemma.

Lemma 5.11. *There are at most 2^{i2^i} cliques in $\mathbb{C}$ of Type B with a unique neighbourhood in D.*

Lemma 5.12. *Given a graph $G = (V, E)$, $D \subseteq V$ such that $|D|$ is the distance to cluster of G, there exists an exponential kernel of size at most $|D| + |D| \cdot 4^{|D|} \cdot (|D| \cdot k + 2^{|D|2^{|D|}} \cdot 2^{|D|})$.*

Proof.

- From Lemma 5.6, we have a bound of 2^i on the size of the clique outside the distance to cluster.
- From Lemma 5.7, we have a bound of $i \cdot k$ on the number of cliques of Type A.
- From Lemma 5.9, we have a bound of 2^i on the number of cliques of Type B, with the same adjacency in D.
- From Lemma 5.11, we have a bound of 2^{i2^i} on the number of cliques of Type B, with a unique neighbourhood in D.

Combining Lemmas 5.7, 5.9 and 5.11, we have a bound of $(i \cdot k + 2^{i2^i} \cdot 2^i)$ on the number of cliques in $\mathbb{C}$. The size of $\mathbb{C}$ will be at most $2^i \cdot (i \cdot k + 2^{i2^i} \cdot 2^i)$. There will be at most $\binom{|D|}{i}$ clique sets in C_i. The size of C_i is upper bounded by $\binom{|D|}{i} \cdot 2^i \cdot (i \cdot k + 2^{i2^i} \cdot 2^i)$. The value of i varies from 1 to $|D|$. Hence, we obtain the following upper bound on the kernel size.

$$
\begin{aligned}
&= |D| + \sum_{i=1}^{|D|} \binom{|D|}{i} \cdot 2^i \cdot \left(i \cdot k + 2^{i2^i} \cdot 2^i\right) \\
&= \mathcal{O}\left(|D| + |D| \cdot 2^{|D|} \cdot 2^{|D|} \cdot \left(|D| \cdot k + 2^{|D|2^{|D|}} \cdot 2^{|D|}\right)\right) \\
&= \mathcal{O}\left(|D| + |D| \cdot 4^{|D|} \cdot \left(|D| \cdot k + 2^{|D|2^{|D|}} \cdot 2^{|D|}\right)\right).
\end{aligned}
$$

$\square$

As there exists an exponential kernel for the problem in terms of the parameters distance to cluster, membership(k); we obtain the following result.

Theorem 5.13. *Given a graph $G = (V, E)$, $D \subseteq V$ such that $|D|$ is distance to cluster of G, the* Minimum Membership Dominating Set *problem is in FPT parameterized by distance to cluster, membership(k).*

6. Linear time algorithm for trees

We analyze the complexity of Minimum Membership Dominating Set on trees and present a linear-time algorithm using dynamic programming. Our approach computes partial solutions in a bottom-up manner, starting from the leaves and progressing to the root.

Consider a tree, $G = (V, E)$ rooted at an arbitrary node r. Let G_u denote the subgraph induced by node u and all its descendants. We use S_u to denote a *minimum membership dominating set* of G_u. At each node $u \in V$, we maintain four boolean variables: $M^+(u), M^-(u), M(u)$ and $M'(u)$. These variables are defined as follows.

- $M^+(u)$ indicates whether there exists an S_u such that $u \in S_u$.
- $M^-(u)$ indicates whether there exists an S_u such that $u \notin S_u$.
- $M(u)$ is true if there exists an S_u, regardless of whether u is included in it.
- $M'(u)$ is true if $M^-(u) = 0$ solely because u is undominated. If $M^-(v) = 1$ for each child node $v \in C(u)$ and $M^-(u) = 0$, this would indicate that $M^-(u) = 0$ only because u is undominated.

We obtain the values of $M^+(u)$, $M^-(u), M(u)$ and $M'(u)$ as follows:

- $M^+(u) = \begin{cases} 1, & \text{if there exists an } S_u \text{ such that } u \in S_u \\ 0, & \text{otherwise.} \end{cases}$
- $M^-(u) = \begin{cases} 1, & \text{if there exists an } S_u \text{ such that } u \notin S_u \\ 0, & \text{otherwise.} \end{cases}$
- $M(u) = M^-(u) \vee M^+(u)$.
- $M'(u) = \begin{cases} 1, & \text{if (1) } M^-(u) = 0 \text{ and} \\ & \text{(2) For each } v \in C(u),\ M^-(v) = 1 \\ 0, & \text{otherwise.} \end{cases}$

The existence of a *minimum membership dominating set* for G can be obtained based on the value of $M(r)$.

Leaf node

For a leaf node u, we set $M^+(u) = 1$ and $M^-(u) = 0$. We obtain $M(u) = 1$. As $M^-(u) = 0$ only because u is undominated, we set $M'(u) = 1$.

Non-leaf node

For each non-leaf node u, let $C(u)$ denote the set consisting of the child nodes of u. We also define three boolean variables: $P(u), Q(u)$ and $R(u)$. $P(u)$ indicates whether there exists an S_u with at most k child nodes of u (say v) with $M^-(v) = 0$. $Q(u)$ indicates whether there exists an S_u with at most $k-1$ child nodes of u (say v) with $M'(v) = 0$. $R(u)$ indicates whether there exists an S_u with no child node of u (say v) having k closed neighbours in S_v. The values of $P(u), Q(u)$ and $R(u)$ are computed as follows.

- $P(u) = \begin{cases} 0, & \text{if } \sum_{v \in C(u)} (1 - M^-(v)) > k \\ 1, & \text{otherwise.} \end{cases}$
- $Q(u) = \begin{cases} 0, & \text{if } \sum_{\substack{v \in C(u) \\ M^-(v)=0}} (1 - M'(v)) > k - 1 \\ 1, & \text{otherwise.} \end{cases}$
- $R(u) = \begin{cases} 0, & \text{if there exists some vertex } v \in C(u) \text{ such that} \\ & ((1 - M^-(v))(1 - M'(v)) + \sum_{\substack{w \in C(v) \\ M^-(w)=0}} (1 - M'(w))) > k - 1 \\ 1, & \text{otherwise.} \end{cases}$

Note: Consider a subgraph G_u. Vertex u must be included in S_u if $M^-(u) = 0$ and $M'(u) = 0$. But, no where in this approach, we will discuss regarding the inclusion of a vertex u in S_u based on $M^-(u)$ and $M'(u)$ values. We simply calculate the values of $M^+(u)$, $M^-(u)$, $M(u)$ and $M'(u)$, for each vertex $u \in V$. Ultimately, the value of $M(r)$ determines whether the graph G admits a *minimum membership dominating set.*

For a non-leaf node, we compute the values of $M^-(u), M'(u), M^+(u)$ and $M(u)$ as follows.

Computing $M^-(u)$

Let $u \notin S_u$ and for a child node $v \in C(u)$, $M^-(v) = 0$. Irrespective of the value of $M'(v)$, v must be a part of S_u. As $u \notin S_u$, if $M^-(v) = 0$ then the vertex v must belong to S_u.

We compute the value of $M^-(u)$ as follows.

$$M^-(u) = \left(\prod_{v \in C(u)} M(v)\right) \cdot \left(\vee_{v \in C(u)} \left(1 - M^-(v)\right)\right) \cdot P(u).$$

Here, $M^-(u) = 1$, if and only if,

- For each child node (say v) of u, $M(v) = 1$.
- Among all the child nodes (say v) of u, $M^-(v) = 0$ for at least one of them. This will make sure that u is dominated.
- Among all the child nodes (say v) of u, $M^-(v) = 0$ for at most k of them.

Computing $M'(u)$

We compute the value of $M'(u)$ as follows.

$$M'(u) = \left(1 - M^-(u)\right) \cdot \left(\prod_{v \in C(u)} M^-(v)\right).$$

Here, $M'(u) = 1$, if and only if,

- $M^-(u) = 0$.
- For each child node (say v) of u, $M^-(v) = 1$.

Computing $M^+(u)$

To start with, we perform the following initial checks.

- We determine whether there exists an S_u such that at most $k-1$ child nodes of u are in S_u (as defined through $Q(u)$).
- For each child node (say v) of u, we check whether there exists an S_v such that v has at most $k-1$ closed neighbours in S_v (as defined through $R(u)$).

If either of these conditions is not satisfied, we set $M^+(u) = 0$ and terminate the computation of $M^+(u)$.

Now, we check whether $M(v) = 1$ for each child node $v \in C(u)$. Suppose there exists a child node v such that $M(v) = 0$. This implies that no *minimum membership dominating set* exists for the subgraph G_v. However, it is still possible that $M(u) = 1$, and we handle this case accordingly.

The only reason for which $M(u) = 1$ despite $M(v) = 0$ is if $M(v) = 0$ solely because v is undominated. If vertex u is included in S_u, it can dominate v. Since u can only dominate v, all other MMDS constraints for the nodes in G_v must still be satisfied. To ensure this, the following conditions must hold:

(1) As v is not a part of S_v; S_{w_i} must exist for each child node w_i of v. Hence, for each child node w_i of v, $M(w_i) = 1$.
(2) In order for v to be undominated, v must not be a part of S_v and all the child nodes of v must not be a part of S_v.

- *v to be not a part of S_v:* for at least one child node of v (say w_i), exactly k child nodes of w_i must be in S_v.
- *No child node of v to be a part of S_v:* for each child node of v (say w_i), either k child nodes of w_i must be in S_v or there must exist some child node of w_i (say w_i^j) such that k nodes from $\{w_i^j$ and child nodes of $w_i^j\}$, must be in S_v.
  In other words, for each node $w_i \in C(v)$, one of the following two must hold:
  - $M'(x) = 0$ for k nodes from $x \in \{w_i^1, w_i^2, \ldots, w_i^y\}$, where w_i^j represents a child node of w_i. **Note:** This condition must be satisfied for at least one child node $w_i \in C(v)$, so that $v \notin S_v$.
  - Let $w_i^{j,l}$ represents a child node of w_i^j. $M'(x) = 0$ for k nodes from $x \in \{w_i^j, w_i^{j,1}, \ldots, w_i^{j,z}\}$.

See Figure 6 for an illustration.

We compute the value of $M^+(u)$ as follows.

$$M^+(u) = Q(u) \cdot R(u) \cdot \prod_{v \in C(u)} W(v)$$

where

$$W(v) = M(v) \vee (X(v) \wedge Y(v) \wedge Z(v))$$

such that

$$X(v) = \prod_{w_i \in C(v)} M(w_i),$$

$$Y(v) = \left\{ \exists_{w_i \in C(v)} \left(\sum_{\substack{w_i^j \in C(w_i) \\ M^-(w_i^j)=0}} \left(1 - M'\left(w_i^j\right)\right) \right) = k?1:0 \right\},$$

$$\begin{aligned} Z(v) = \Bigg(\Bigg[\sum_{w_i \in C(v)} \Bigg\{ \Bigg(\Bigg(\Bigg(\sum_{\substack{w_i^j \in C(w_i) \\ M^-(w_i^j)=0}} \left(1 - M'\left(w_i^j\right)\right) \Bigg) = k?1:0 \Bigg) \vee \Bigg(\exists_{\substack{w_i^j \in C(w_i) \\ M^-(w_i^j)=0}} \Bigg(\left(1 - M'\left(w_i^j\right)\right) \\ + \sum_{\substack{w_i^{j,l} \in C(w_i^j) \\ M^-(w_i^{j,l})=0}} \left(1 - M'\left(w_i^{j,l}\right)\right) \Bigg) = k?1:0 \Bigg) \Bigg) \Bigg\} \Bigg] = |C(v)|?1:0 \Bigg). \end{aligned}$$

$X(v) = 1$, if and only if, for each child node $w_i \in C(v)$, $M(w_i) = 1$.

$Y(v) = 1$, if and only if, there exists some child node $w_i \in C(v)$ with k of the child nodes of w_i in S_v. This is to make sure that $v \notin S_v$.

$Z(v) = 1$, if and only if, for each child node $w_i \in C(v)$, either

(1) k of the child nodes of w_i are in S_v or
(2) there exists some node $w_i^j \in C(w_i)$ such that k of the child nodes of w_i^j (including w_i^j) are in S_v.

This is to make sure that no child node of v (say w_i) belongs to S_v.

Here, $M^+(u) = 1$, if and only if, $Q(u) = 1$, $R(u) = 1$ and for each child node (say v) of u, $W(v) = 1$.

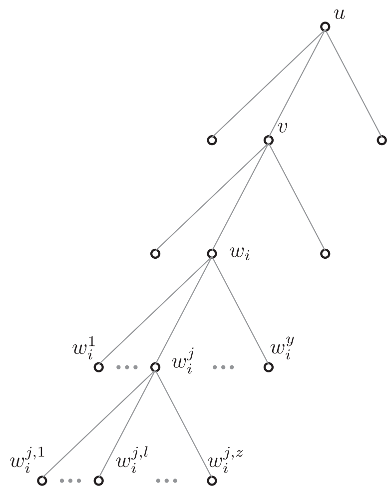

FIGURE 6. Handling the scenario of $M(u) = 1$ when $M(v) = 0$.

Computing $M(u)$

$$M(u) = M^-(u) \vee M^+(u).$$

There exists a *minimum membership dominating set* for G, if and only if $M(r) = 1$. Hence, we obtain the following theorem.

Theorem 6.1. *The* MINIMUM MEMBERSHIP DOMINATING SET *problem on trees is linear-time solvable.*

7. CONCLUSION

In this work, we have studied a variant of the DOMINATING SET, namely the MINIMUM MEMBERSHIP DOMINATING SET problem. For split graphs, we present an exact algorithm with a running time of $\mathcal{O}^*(1.5719^n)$. For bipartite graphs, we show that, assuming ETH, no subexponential-time algorithm exists for solving the problem. Additionally, we prove that the problem remains NP-complete for graphs with maximum degree $\Delta = k + 2$, when $k \geq 5$, even for bipartite graphs. On the positive front, we demonstrate that the problem is fixed-parameter tractable with respect to the twin cover number, as well as the combined parameter distance to cluster and membership(k). Finally, we provide a linear-time algorithm for solving the problem for trees.

As part of future work, one could look to obtain non-trivial exact algorithms for bipartite graphs (or even for planar bipartite graphs) and general graphs. The lower bound for the problem on planar bipartite graphs (or on planar graphs) is also an interesting open question. The complexity of the problem on bipartite graphs with maximum degree, $\Delta = k + 1$, is yet to be resolved. Similarly, the complexity of the problem on interval graphs, block graphs, and cographs is unknown. Exploring the parameterized complexity with respect to the feedback vertex set number offers a promising research direction. Furthermore, the existence of polynomial kernels for parameters vertex cover number, feeback edge set number and distance to clique remains an open problem.

DATA AVAILABILITY STATEMENT

No new data/codes were created or analyzed in this study.

References

[1] F.N. Abu-Khzam, An improved exact algorithm for minimum dominating set in chordal graphs. *Inf. Process. Lett.* **174** (2022) 106206.

[2] A. Agrawal, P. Choudhary, N.S. Narayanaswamy, K.K. Nisha and V. Ramamoorthi, Parameterized complexity of minimum membership dominating set. *Algorithmica* **85** (2023) 3430–3452.

[3] M. Alambardar Meybodi, F.V. Fomin, A.E. Mouawad and F. Panolan, On the parameterized complexity of $[1, j]$-domination problems. *Theor. Comput. Sci.* **804** (2020) 207–218.

[4] J. Alber and R. Niedermeier, Improved tree decomposition based algorithms for domination-like problems, in Proceedings of the 5th Latin American Symposium on Theoretical Informatics, LATIN '02. Springer-Verlag, Berlin, Heidelberg (2002) 613–628.

[5] A.A. Bertossi, Dominating sets for split and bipartite graphs. *Inf. Process. Lett.* **19** (1984) 37–40.

[6] M. Chellali, T.W. Haynes, S.T. Hedetniemi and A. McRae, $[1, 2]$-sets in graphs. *Discrete Appl. Math.* **161** (2013) 2885–2893.

[7] M. Cygan, F.V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk and S. Saurabh, Parameterized Algorithms. Springer, Switzerland (2015).

[8] F.V. Fomin and D. Kratsch, Exact Exponential Algorithms. Springer (2010).

[9] F.V. Fomin, F. Grandoni and D. Kratsch, A measure & conquer approach for the analysis of exact algorithms. *J. ACM (JACM)* **56** (2009) 1–32.

[10] F.V. Fomin, D. Kratsch and G.J. Woeginger, Exact (exponential) algorithms for the dominating set problem, in Graph-Theoretic Concepts in Computer Science, edited by J. Hromkovič, M. Nagl and B. Westfechtel. Springer Berlin Heidelberg, Berlin, Heidelberg (2005) 245–256.

[11] R. Ganian, Improving vertex cover as a graph parameter. *Discrete Math. Theor. Comput. Sci.* **17** (2015) 77–100.

[12] A.K. Goharshady, M.R. Hooshmandasl and M.A. Meybodi, $[1, 2]$-sets and $[1, 2]$-total sets in trees with algorithms. *Discrete Appl. Math.* **198** (2016) 136–146.

[13] Y. Iwata, A faster algorithm for dominating set analyzed by the potential method, in Parameterized and Exact Computation, edited by D. Marx and P. Rossmanith. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 41–54.

[14] R.M. Karp, Reducibility among Combinatorial Problems. Springer US, Boston, MA (1972).

[15] D. Karthika, R. Muthucumaraswamy, M. Bentert, S. Bhyravarapu, S. Saurabh and S. Seetharaman, On the complexity of minimum membership dominating set, in International Conference on Current Trends in Theory and Practice of Computer Science. Springer (2025) 94–107.

[16] T. Kikuno, N. Yoshida and Y. Kakuda, The *np*-completeness of the dominating set problem in cubic planer graphs. *IEICE Trans. (1976–1990)* **63** (1980) 443–444.

[17] F. Kuhn, P. von Rickenbach, R. Wattenhofer, E. Welzl and A. Zollinger, Interference in cellular networks: the minimum membership set cover problem, in Computing and Combinatorics, edited by L. Wang. Springer Berlin Heidelberg, Berlin, Heidelberg (2005) 188–198.

[18] M. Liedloff, Finding a dominating set on bipartite graphs. *Inf. Process. Lett.* **107** (2008) 154–157.

[19] N.S. Narayanaswamy, S.M. Dhannya and C. Ramya, Minimum membership hitting sets of axis parallel segments, in Computing and Combinatorics, edited by L. Wang and D. Zhu. Springer International Publishing, Cham (2018) 638–649.

[20] S. Porschen, T. Schmidt, E. Speckenmeyer and A. Wotzlaw, XSAT and NAE-SAT of linear CNF classes. *Discrete Appl. Math.* **167** (2014) 1–14.

[21] T.J. Schaefer, The complexity of satisfiability problems, in Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, STOC '78. Association for Computing Machinery, New York, NY, USA (1978) 216–226.

[22] J.M.M. van Rooij and H.L. Bodlaender, Exact algorithms for dominating set. *Discrete Appl. Math.* **159** (2011) 2147–2164.

[23] J.M.M. van Rooij, H.L. Bodlaender and P. Rossmanith, Dynamic programming on tree decompositions using generalised fast subset convolution, in Algorithms – ESA 2009, edited by A. Fiat and P. Sanders. Springer Berlin Heidelberg, Berlin, Heidelberg (2009) 566–577.

[24] D.B. West, Introduction to Graph Theory, 2 edition. Pearson, Chennai (2015).