New Algorithms for Mixed Dominating Set

Louis Dublois
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, Paris, France
louis.dublois@lamsade.dauphine.fr

Michael Lampis
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, Paris, France
michail.lampis@lamsade.dauphine.fr

Vangelis Th. Paschos
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, Paris, France
paschos@lamsade.dauphine.fr

Abstract

A mixed dominating set is a set of vertices and edges that dominates all vertices and edges of a graph. We study the complexity of exact and parameterized algorithms for MDS, resolving some open questions. In particular, we settle the problem's complexity parameterized by treewidth and pathwidth by giving an algorithm running in time $O^*(5^{tw})$ (improving the current best $O^*(6^{tw})$), and a lower bound showing that our algorithm cannot be improved under the SETH, even if parameterized by pathwidth (improving a lower bound of $O^*((2-\varepsilon)^{pw})$). Furthermore, by using a simple but so far overlooked observation on the structure of minimal solutions, we obtain branching algorithms which improve the best known FPT algorithm for this problem, from $O^*(4.172^k)$ to $O^*(3.510^k)$, and the best known exact algorithm, from $O^*(2^n)$ and exponential space, to $O^*(1.912^n)$ and polynomial space.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases FPT Algorithms, Exact Algorithms, Mixed Domination

Digital Object Identifier 10.4230/LIPIcs.IPEC.2020.9

1 Introduction

Domination problems in graphs are one of the most well-studied topics in theoretical computer science. In this paper we study a variant called Mixed Dominating Set: we are given a graph $G = (V, E)$ and are asked to select $D \subseteq V$ and $M \subseteq E$ such that $|D \cup M|$ is minimized and the set $D \cup M$ dominates $V \cup E$, where a vertex dominates itself, its neighbors, and its incident edges and an edge dominates itself, its endpoints, and all edges with which it shares an endpoint.

Mixed Dominating Set is a natural variation of domination in graphs as it can be seen as a *mix* between four standard problems: Dominating Set, where vertices dominate vertices; Edge Dominating Set, where edges dominate edges; Vertex Cover, where vertices dominate edges; and Edge Cover, where edges dominate vertices. In Mixed Dominating Set we are asked to select vertices *and* edges in a way that dominates all vertices *and* edges. As only the last of these four problems is in P, it is not surprising that Mixed Dominating Set is NP-hard. We are therefore motivated to study approximation, exponential-time and parameterized algorithms for this problem, and indeed this has been the topic of several recent papers. On the approximation algorithms side, the problem is well-understood: Hatami [9] gave a 2-approximation algorithm, while more recently Dudycz et al. [5] showed that (under the UGC) no algorithm can achieve a ratio better than 2 for Edge Dominating Set. As we explain (Proposition 1) this hardness result easily carries over to Mixed Dominating Set, thus essentially settling the problem's approximability. Hence, in this paper we focus on parameterized and exact algorithms.

Mixed Dominating Set has recently been the focus of several works in this context. With respect to the natural parameter (the size k of the solution), an $O^*(7.465^k)$[1] algorithm was given by Jain et al. [12], more recently improved to $O^*(4.172^k)$ by Xiao and Sheng [26]. With respect to treewidth and pathwidth, Jain et al. gave algorithms running in $O^*(6^{tw})$ time and $O^*(5^{pw})$ time, improving upon the $O^*(3^{tw^2})$ time algorithm of [24]. Furthermore, Jain et al. showed that no algorithm can solve the problem in $O^*((2-\varepsilon)^{pw})$ time under the Set Cover Conjecture. These works observed that it is safe to assume that the optimal solution has a nice structure: the selected edges form a matching whose endpoints are disjoint from the set of selected vertices. This observation immediately gives an $O^*(3^n)$ algorithm for the problem, which was recently improved to $O^*(2^n)$ by Madathil et al. [18] by using a dynamic programming approach, which requires $O^*(2^n)$ space.

Our results

The state of the art summarized above motivates two basic questions: first, can the gap in the complexity of the problem for treewidth and pathwidth and the gap between the lower and upper bound for these parameters be closed, as explicitly asked in [12]; second, can we solve this problem faster than the natural $O^*(2^n)$ barrier? We answer these questions and along the way obtain an improved FPT algorithm for parameter k. Specifically we show:

(i) Mixed Dominating Set can be solved in $O^*(5^{tw})$ time. Somewhat surprisingly, this result is obtained by combining observations that exist in the literature: the equivalence of Mixed Dominating Set to Distance-2-Dominating Set [18]; and the algorithm of Borradaile and Le for this problem [3].

(ii) Mixed Dominating Set cannot be solved in time $O^*((5-\varepsilon)^{pw})$, under the SETH. This is our main result on this front, and shows that our algorithm for treewidth and the algorithm of [12] for pathwidth are optimal.

(iii) Mixed Dominating Set can be solved in time $O^*(1.912^n)$ and $O^*(3.510^k)$, in both cases using polynomial space. In order to obtain these algorithms we refine the notion of *nice mixed dominating set* which was used in previous algorithms. In particular, we show that a mixed dominating set with the minimum number of vertices has the property that any selected vertex has at least two *private* neighbors. This allows us to speed up branching on low-degree vertices.

Other related work

The notion of Mixed Dominating Set was first introduced in 1977 by Alavi. et al [1], and has been studied extensively in graph theory [2,6,21,23]. See the chapter in [10] for a survey on the Mixed Dominating Set problem. The computational complexity of Mixed Dominating Set was first studied in 1993 by Majumbar [19], where he showed that the problem is NP-complete. The problem remains NP-complete on split graphs [27] and on planar bipartite graphs of maximum degree 4 [20]. Majumbar [19], Lan and Chang [16], Rajaati et al. [25] and Madathil et al. [18] showed that the problem is polynomial-time solvable on trees, cacti, generalized series-parallel graphs and proper interval graphs, respectively.

[1] O^* notation suppresses polynomial factors in the input size.

2 Preliminaries

We assume familiarity with the basics of parameterized complexity (e.g. treewidth, pathwidth, and the SETH), as given in [4]. Let $G = (V, E)$ be a graph with $|V| = n$ vertices and $|E| = m$ edges. For $u \in V$, $N(u)$ denotes the set of neighbors of u, $d(u) = |N(u)|$ and $N[u] = N(u) \cup \{u\}$. For $U \subseteq V$ and $u \in V$, we note $N_U(u) = N(u) \cap U$ and use $d_U(u)$ to denote $|N_U(u)|$. Furthermore, for $U \subseteq V$ we denote $N(U) = \cup_{u \in U} N(u)$. For an edge set E', we use $V(E')$ to denote the set of endpoints of E'. For $V' \subseteq V$, we use $G[V']$ to denote the subgraph of G induced by V'. A *mixed dominating set* of a graph $G = (V, E)$ is a set of vertices $D \subseteq V$ and edges $M \subseteq E$ such that (i) all vertices of $V \setminus (D \cup V(M))$ have a neighbor in D (ii) all edges of $E \setminus M$ have an endpoint in $D \cup V(M)$.

We note that the minimization problem Mixed Dominating Set is harder than the more well-studied Edge Dominating Set (EDS) problem, in a way that preserves most parameters and the size of the optimal solution. Hence, essentially all hardness results for the latter problem, such as its inapproximability [5] or its W[1]-hardness for clique-width [7], carry over to Mixed Dominating Set.

▶ **Proposition 1.** *There is an approximation and parameter-preserving reduction from Edge Dominating Set to Mixed Dominating Set.*

Proof. Given an instance $G = (V, E)$ of EDS we seek a set M of k edges such that all edges have an endpoint in $V(M)$. We add a new vertex u connected to all of V and attach to u $|V| + 2$ leaves. The new graph has a mixed dominating set of size $k + 1$ if and only if G has an edge dominating set of size k. ◀

We now define a restricted notion of mixed dominating set.

▶ **Definition 2.** *A* nice *mixed dominating set of a graph $G = (V, E)$ is a mixed dominating set $D \cup M$ which satisfies the following: (i) $D \cap V(M) = \emptyset$; (ii) M is a matching; (iii) for all $u \in D$ there exist at least two private neighbors of u, that is, two vertices $v_1, v_2 \in V \setminus (D \cup V(M))$ with $N(v_1) \cap D = N(v_2) \cap D = \{u\}$.*

We note that a mixed dominating set that satisfies the first two properties of Definition 2 was called *special mds* in [18]. The notion of nice mds was implicit also in the algorithms of [12, 26], with the key difference that these algorithms do not use the fact that every vertex of D must have at least two *private* neighbors, that is, two neighbors which are dominated only by this vertex.

Let us now prove that restricting ourselves to nice solutions does not change the value of the optimal. The idea behind the proof is to reuse the arguments of [18] to obtain an optimal solution satisfying the first two properties; and then while there exists $u \in D$ with at most one private neighbor, we replace it by an edge while maintaining a valid solution satisfying the first two properties.

▶ **Lemma 3.** *For any graph $G = (V, E)$ without isolated vertices, G has a mixed dominating set $D \cup M$ of size at most k if and only if G has a nice mixed dominating set $D' \cup M'$ of size at most k.*

Proof. One direction is trivial, since any nice mixed dominating set is also by definition a mixed dominating set. For the other direction, we first recall that it was shown in [18] that if a graph has a mixed dominating set of size k, then it also has such a set that satisfies the first two conditions of Definition 2. Suppose then that $D \cup M$ is such that $D \cap V(M) = \emptyset$ and M is a matching.

We will now edit this solution so that we obtain the missing desired properties, namely the fact that all vertices of D have two private neighbors. Our transformations will be applicable as long as there exists a vertex $u \in D$ without two private neighbors, and will either decrease the size of the solution, or decrease the size of D, while maintaining a valid solution satisfying the first two properties of Definition 2. As a result, applying these transformations at most n times yields a nice mixed dominating set.

Let $I = V \setminus (D \cup V(M))$. If there exists $u \in D$ with exactly one private neighbor, let v be this private neighbor. We set $D' := D \setminus \{u\}$ and $M' := M \cup \{(u, v)\}$ to obtain another solution. This solution is valid because $N(u) \setminus \{v\}$ is dominated by $(D \cup M) \setminus \{u\}$, otherwise u would have more than one private neighbor.

Let us now consider a vertex $u \in D$ with no private neighbor. Note that for such a vertex u, its neighborhood (which is non-empty, since G has no isolated vertices) is dominated by $(D \cup M) \setminus \{u\}$, because otherwise u would have at least one private neighbor. If there exists $v \in N(u) \cap I$, set $D' := D \setminus \{u\}$ and $M' := M \cup \{(u, v)\}$ to obtain another feasible solution.

Now consider a vertex $u \in D$ with no private neighbor for which $N(u) \cap I = \emptyset$. We have $N(u) \subseteq D \cup V(M)$, which implies that the neighborhood of u and all the edges incident on u are dominated by $(D \cup M) \setminus \{u\}$. If there exists $v \in N(u) \cap D$, remove u from the solution to get a better solution. Now consider a vertex $u \in D$ with no private neighbor for which $N(u) \subseteq V(M)$. If there exists $v \in N(u)$ with $(v, w) \in M$ such that there exists $z \in N(w) \cap I$, set $D' := D \setminus \{u\}$ and $M' := (M \setminus \{(v, w)\}) \cup \{(u, v), (w, z)\}$ to obtain another feasible solution. If for all $v \in N(u)$ with $(v, w) \in M$, we have $N(w) \subseteq D \cap V(M)$, pick such a vertex v and set $D' := (D \setminus \{u\}) \cup \{v\}$ and $M' := M \setminus \{(v, w)\}$ to get a better solution. We repeat these modifications until we obtain the claimed solution. ◀

In the remainder, when considering a mixed dominating set $D \cup M$ of a graph $G = (V, E)$, we will associate with it the partition $V = D \cup P \cup I$ where $P = V(M)$ and $I = V \setminus (D \cup P)$. We will call this a nice mds partition. It is not hard to see that the following properties follow from the definition: (i) $G[P]$ has a perfect matching (ii) I is an independent set (iii) D dominates I (iv) each $u \in D$ has two private neighbors $v_1, v_2 \in I$, that is, $N(v_1) \cap D = N(v_2) \cap D = \{u\}$.

We also note the following useful relation.

▶ **Lemma 4.** *For any graph $G = (V, E)$ and any nice mds partition $V = D \cup P \cup I$ of G, there exists a minimal vertex cover C of G such that $D \subseteq C \subseteq D \cup P$.*

Proof. Since I is an independent set of G, $D \cup P$ is a vertex cover of G and hence contains some minimal vertex cover. We claim that any such minimal vertex cover $C \subseteq D \cup P$ satisfies $D \subseteq C$. Indeed, for each $u \in D$ there exist two private neighbors $v_1, v_2 \notin D \cup P$. Hence, if $u \notin C$, the edge (u, v_1) is not covered, contradiction. ◀

3 Treewidth

We begin with an algorithm for Mixed Dominating Set running in time $O^*(5^{tw})$. We rely on three ingredients: (i) the fact that Mixed Dominating Set on G is equivalent to Distance-2-Dominating Set on the incidence graph of G [18] ; (ii) the standard fact that the incidence graph of G has the same treewidth as G ; (iii) and an $O^*(5^{tw})$ algorithm (from [3]) for Distance-2-Dominating Set.

▶ **Theorem 5.** *There is an $O^*(5^{tw})$-time algorithm for Mixed Dominating Set in graphs of treewidth tw.*

The main result of this section is a lower bound matching Theorem 5. We prove that, under SETH, for all $\varepsilon > 0$, there is no algorithm for Mixed Dominating Set with complexity $O^*((5-\varepsilon)^{pw})$. The starting point of our reduction is the problem q-CSP-5 [15]. In this problem we are given a Constraint Satisfaction (CSP) instance with n variables and m constraints. The variables take values in a set of size 5, say $\{0,1,2,3,4\}$. Each constraint involves at most q variables and is given as a list of acceptable assignments for these variables. The following result was shown in [15] to be a natural consequence of the SETH.

▶ **Lemma 6** (Theorem 2 of [15]). *If the SETH is true, then for all $\varepsilon > 0$, there exists a q such that n-variable q-CSP-5 cannot be solved in time $O^*((5-\varepsilon)^n)$.*

Note that in [15] it was shown that for any alphabet size B, q-CSP-B cannot be solved in time $O^*((B-\varepsilon)^n)$ under the SETH, but for our purposes only the case $B = 5$ is relevant for two reasons: because this corresponds to the base of our target lower bound ; and because in our construction we will represent the $B = 5$ possible values for a variable with a path of five vertices in which there exists exactly five different ways of selecting one vertex and one edge among these five vertices. Our plan is therefore to produce a polynomial-time reduction which, given a q-CSP-5 instance with n variables, produces an equivalent Mixed Dominating Set instance whose pathwidth is at most $n + O(1)$. Then, the existence of an algorithm for the latter problem running faster than $O^*((5-\varepsilon)^{pw})$ would give an $O^*((5-\varepsilon)^n)$ algorithm for q-CSP-5, contradicting the SETH.

Before giving the details of our reduction let us sketch the basic ideas, which follow the pattern of other SETH-based lower bounds which have appeared in the literature [8,11,13,14,17]. The constructed graph consists of a main selection part of n paths of length $5m$, divided into m sections. Each path corresponds to a variable and each section to a constraint. The idea is that the optimal solution will follow for each path a basic pattern of selecting one vertex and one edge among the first five vertices and then repeat this pattern throughout the path (see Figure 1). There are 5 natural ways to do this, so this can represent all assignments to the q-CSP-5 instance. We will then add verification gadgets to each section, connected only to the vertices of that section that represent variables appearing in the corresponding constraint (thus keeping the pathwidth under control), in order to check that the selected assignment satisfies the constraint.

The main difficulty in completing the proof is showing that the optimal solution has the desired form, and in particular, that the pattern that is selected for a variable is kept constant throughout the construction. This is in general not possible to prove, but using a technique introduced in [17], we work around this difficulty by making polynomially many copies of our construction, gluing them together, and arguing that a large enough consistent copy must exist.

Construction

We are given a q-CSP-5 instance φ with n variables $x_1, \ldots, x_n$ taking values over the set $\{0,1,2,3,4\}$, and m constraints $c_0, \ldots, c_{m-1}$. For each constraint we are given a set of at most q variables which are involved in this constraint and a list of satisfying assignments for these variables. Without loss of generality, we make the following assumptions: (i) each constraint involves exactly q variables, because if it has fewer variables, we can add to it new variables and augment the list of satisfying assignments so that the value of the new variables is irrelevant (ii) all constraints have lists of satisfying assignments of size $C = 5^q - 1$; note that this is an upper bound on the size of the list of satisfying assignments, and for each constraint which has fewer we add several copies of one of its satisfying assignments to its list (so the list may repeat an assignment). We define two "large" numbers $F = (3n+1)(2n+1)$ and $A = 20$ and we set our budget to be $k = 8AFmn + 2Fmn + 2Fmq(C-1) + n + 1$.

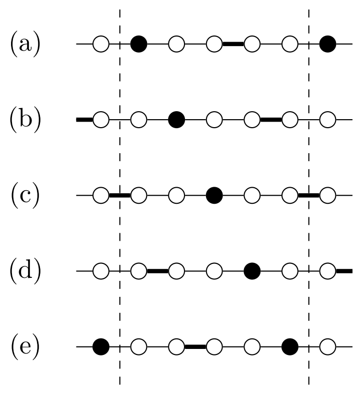

Figure 1 Main part of the construction with the five possible configurations. Filled vertices are in D, thick edges are in M.

We now construct our graph as follows:

1. We construct a vertex s and attach to it two leaves s_1, s_2.
2. For $i \in \{1, \dots, n\}$ we construct a path on $5Fm$ vertices: the vertices are labeled $u_{i,j}$, for $j \in \{0, 1, \dots, 5Fm - 1\}$ and for each i, j the vertex $u_{i,j}$ is connected to $u_{i,j+1}$. We call these paths the *main* part of our construction.
3. For each $j \in \{0, 1, \dots, Fm - 1\}$, let $j' = j \bmod m$. We construct a checker gadget H_j as follows (see Figure 2):

   a. For each satisfying assignment σ in the list of the constraint $c_{j'}$, we construct an independent set $Z_{\sigma,j}$ of size $2q$ (therefore, C such independent sets). The $2q$ vertices are partitioned so that for each of the q variables involved in $c_{j'}$ we reserve two vertices. In particular, if x_i is involved in $c_{j'}$ we denote by $z^1_{\sigma,j,i}, z^2_{\sigma,j,i}$ its two reserved vertices in $Z_{\sigma,j}$.
   b. For each $i \in \{1, \dots, n\}$ such that x_i is involved in $c_{j'}$, for each satisfying assignment σ in the list of $c_{j'}$, if σ sets x_i to value $\alpha \in \{0, 1, 2, 3, 4\}$ we add the following edges:
      i. $(u_{i,5j+\alpha}, z^1_{\sigma,j,i})$ and $(u_{i,5j+\alpha}, z^2_{\sigma,j,i})$.
      ii. Let $\beta = (\alpha + 2) \bmod 5$ and $\gamma = (\alpha + 3) \bmod 5$. We add the edges $(u_{i,5j+\beta}, z^1_{\sigma,j,i})$ and $(u_{i,5j+\gamma}, z^2_{\sigma,j,i})$.
   c. For all assignments $\sigma \neq \sigma'$ of $c_{j'}$, add all edges between $Z_{\sigma,j}$ and $Z_{\sigma',j}$.
   d. We construct an independent set W_j of size $2q(C - 1)$
   e. Add all edges between W_j and $Z_{\sigma,j}$, for all assignments σ of $c_{j'}$.
   f. For each $w \in W_j$, we construct an independent set of size $2k + 1$ whose vertices are all connected to w and to s.

4. We define the consistency gadget $Q_{i,j}$, for $i \in \{1, \dots, n\}$ and $j \in \{0, \dots, Fm - 1\}$ which consists of (see Figure 2):

   a. An independent set of size 8 denoted $A_{i,j}$.
   b. Five independent sets of size 2 each, denoted $B_{i,j,0}, B_{i,j,1}, \dots, B_{i,j,4}$.
   c. For each $\ell, \ell' \in \{0, \dots, 4\}$ with $\ell \neq \ell'$ all edges from $B_{i,j,\ell}$ to $B_{i,j,\ell'}$.
   d. For each $\ell \in \{0, \dots, 4\}$ all possible edges from $B_{i,j,\ell}$ to $A_{i,j}$.
   e. For each $a \in A_{i,j}$, $2k + 1$ vertices connected to a and to s.

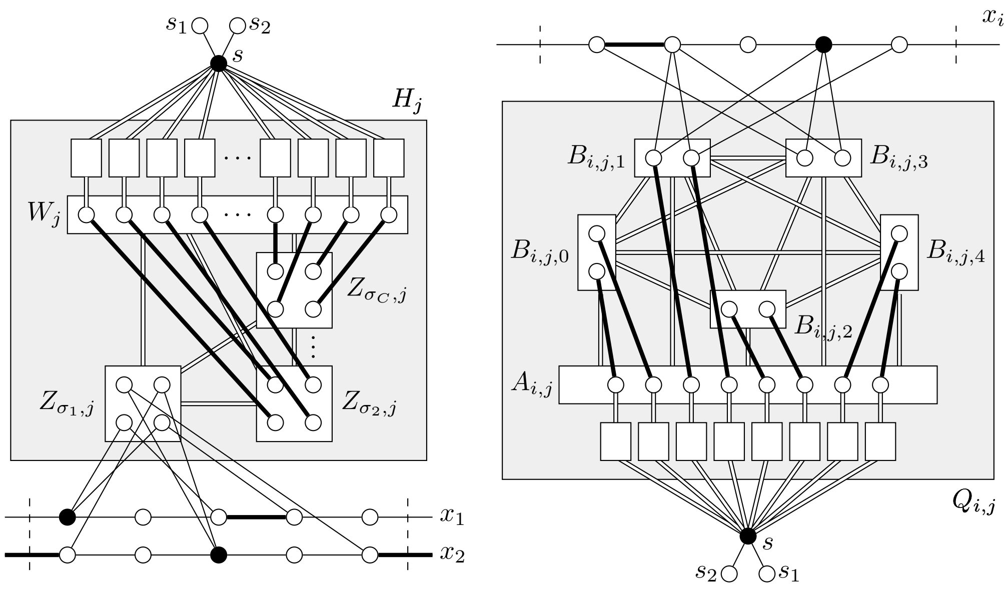

Figure 2 (Double edges between two sets of vertices represent all edges between the two sets.) Left: Checker gadget H_j connected to the main part. Here we have considered an instance where the clause $c_{j'}$ has only two variables, x_1 and x_2. Moreover, only the independent set $Z_{\sigma_1,j}$ is shown connected to the main part. The possible assignment σ_1 of $c_{j'}$ is $(x_1 = 0, x_2 = 2)$. We have supposed that this assignment is satisfiable, and we have have marked the corresponding mixed dominating set: filled vertices are in D, thick edges are in M. Right: Checker gadget $Q_{i,j}$ connected to the main part, that is to the path corresponding to the variable x_i. Only the independent sets $B_{i,j,1}$ and $B_{i,j,3}$ are shown connected to the main part. We have supposed that the assignment $(x_i = 3)$ is satisfiable, and we have marked the corresponding mixed dominating set: filled vertices are in D, thick edges are in M.

f. For each $\ell \in \{0, \dots, 4\}$ both vertices of $B_{i,j,\ell}$ are connected to $u_{i,5j+\ell}$.

g. For each $\ell \in \{0, \dots, 4\}$ let $\ell' = (\ell + 2) \bmod 5$ and $\ell'' = (\ell + 3) \bmod 5$. One vertex of $B_{i,j,\ell}$ is connected to $u_{i,5j+\ell'}$ and the other to $u_{i,5j+\ell''}$.

5. For each $i \in \{1, \dots, n\}$ and $j \in \{0, \dots, Fm - 1\}$ construct A copies of the gadget $Q_{i,j}$ and connect them to the main part as described above.

This completes the construction. The target mds size is k, as defined above. We now argue that the reduction is correct and G has the desired pathwidth.

▶ **Lemma 7.** *If φ is satisfiable, then there exists an mds in G of size at most k.*

Proof. Assume that φ admits some satisfying assignment $\rho : \{x_1, \dots, x_n\} \to \{0, 1, 2, 3, 4\}$. We construct a solution as follows:

1. For each $i \in \{1, \dots, n\}$ let $\alpha = \rho(x_i)$. For each $j \in \{0, \dots, Fm - 1\}$, we select in the dominating set the vertex $u_{i,5j+\alpha}$.
2. Let U' be the set of vertices $u_{i,j}$ of the main part which were not selected in the previous step and which do not have a neighbor selected in the previous step. We add to the solution all edges of a maximum matching of $G[U']$, as well as all vertices of U' left unmatched by this matching.

3. For each $j \in \{0, \dots, Fm-1\}$, G contains a gadget H_j. Consider the constraint $c_{j'}$ for $j' = j \bmod m$. Let σ be an assignment in the list of $c_{j'}$ that agrees with ρ (such a σ must exist, since the constraint is satisfied by ρ). We add to the solution the edges of a perfect matching from W_j to $\bigcup_{\sigma' \neq \sigma} Z_{\sigma',j}$.
4. For each $j \in \{0, \dots, Fm-1\}$ and $i \in \{1, \dots, n\}$ we have added to the graph A copies of the consistency gadget $Q_{i,j}$. For each copy we add to the solution a perfect matching from $A_{i,j}$ to $\bigcup_{\ell \neq \rho(x_i)} B_{i,j,\ell}$.
5. We set $s \in D$.

Let us first argue why this solution has size at most k. In the first step we select Fnm vertices. In the second step we select at most $Fnm + n$ elements. To see this, note that if $u_{i,j}$ is taken in the previous step, then $u_{i,j+5}$ is also taken (assuming $j + 5 < 5Fm$), which leaves two adjacent vertices $(u_{i,j+2}, u_{i,j+3})$. These vertices will be matched in $G[U']$ and in our solution. Note that, for a variable x_i, if $\rho(x_i) \neq 2$, then at most one vertex is left unmatched by the matching taken, so the cost for this variable is at most $Fm + 1$. If $\rho(x_i) = 2$, then at most two vertices are left matched by the matching taken, so the cost for this variable is at most $(Fm - 1) + 2$. Furthermore, for each H_j we select $|W_j| = 2q(C-1)$ edges. For each copy of $Q_{i,j}$ we select 8 edges, for a total cost of $8AFmn$. Taking into account s, the total cost is at most $Fnm + n + Fnm + 2Fmq(C-1) + 8AFmn + 1 = k$.

Let us argue why the solution is feasible. First, all vertices $u_{i,j}$ and all edges connecting them to each other are clearly dominated by the first two steps of our selection. Second, for each H_j, the vertex s together with the endpoints of selected edges form a vertex cover of H_j, so all internal edges are dominated. Furthermore, s dominates all vertices which are not endpoints of our solution, except $Z_{\sigma,j}$, where σ is the selected assignment of $c_{j'}$, with $j' = j \bmod m$. We then need to argue that the vertices of $Z_{\sigma,j}$ and the edges connecting it to the main part are covered.

Recall that the $2q$ vertices of $Z_{\sigma,j}$ are partitioned into pairs, with each pair $z^1_{\sigma,j,i}, z^2_{\sigma,j,i}$ reserved for the variable x_i involved in $c_{j'}$. We now claim that $z^1_{\sigma,j,i}, z^2_{\sigma,j,i}$ are dominated by our solution, since we have selected the vertex $u_{i,5j+\alpha}$, where $\alpha = \rho(x_i)$. Furthermore, $u_{i,5j+\beta}, u_{i,5j+\gamma}$, where $\beta = (a+2) \bmod m$, $\gamma = (a+3) \bmod m$, belong in U' and therefore the edges incident to them are covered. Finally, to see that the $Q_{i,j}$ gadgets are covered, observe that for each such gadget only 2 vertices of some $B_{i,j,\ell}$ are not in P. The common neighbor of these vertices is in D, and their other neighbors in the main part are in P. ◀

The idea of the proof of the next Lemma is the following: by partitioning the graph into different parts and lower bound the cost of these parts, we prove that if a mixed dominating set in G has not the same form as in Lemma 7 in a sufficiently large copy, then it has size strictly greater than k, enabling us to produce a satisfiable assignment for φ using the mixed dominating set which has the desired form.

▶ **Lemma 8.** *If there exists a mixed dominating set in G of size at most k, then φ is satisfiable.*

Proof. Suppose that we are given, without loss of generality (Lemma 3), a nice mixed dominating set of G of minimum cost. We therefore have a partition of $V(G)$ into $V = D \cup P \cup I$, and a perfect matching M of $G[P]$. Before proceeding, let us define for a set $S \subseteq V(G)$ its *cost* as $\text{cost}(S) = |S \cap D| + \frac{|S \cap P|}{2}$. Clearly, $\text{cost}(V(G)) \leq k$ and for disjoint sets S_1, S_2 we have $\text{cost}(S_1 \cup S_2) = \text{cost}(S_1) + \text{cost}(S_2)$. Our strategy will therefore be to partition V into different parts and lower bound their cost.

First, we give some notations. Consider some $j \in \{0, \dots, Fm-1\}$ and $i \in \{1, \dots, n\}$: recall that we have constructed A copies of the gadget $Q_{i,j}$, call them $Q^1_{i,j}, \dots, Q^A_{i,j}$; also let $S_{i,j} = \{u_{i,5j}, u_{i,5j+1}, \dots, u_{i,5j+4}\}$. Now, for some $j \in \{0, \dots, Fm-1\}$, let $S_j = H_j \cup \bigcup_{i\in\{1,\dots,n\}} \left(S_{i,j} \cup \bigcup_{r\in\{1,\dots,A\}} Q^r_{i,j}\right)$.

▷ Claim 9. $\text{cost}(S_j) \geq 2q(C-1) + 2n + 8An$.

Proof. We begin with some easy observations. First, it must be the case that $s \in D$. If not, either s_1 or s_2 are in D, which contradicts the niceness of the solution.

Consider some $j \in \{0, \dots, Fm-1\}$ and $i \in \{1, \dots, n\}$. We will say that, for $1 \leq r \leq A$, $Q^r_{i,j}$ is *normal* if we have the following: $Q^r_{i,j} \cap D = \emptyset$ and there exists $\ell \in \{0, \dots, 4\}$ such that $Q^r_{i,j} \cap P = A_{i,j} \cup \bigcup_{\ell' \neq \ell} B_{i,j,\ell'}$. In other words, $Q^r_{i,j}$ is normal if locally the solution has the form described in Lemma 7.

We now observe that for all i, j, r we have $\text{cost}(Q^r_{i,j}) \geq 8$. To see this, observe that if there exists $a \in A_{i,j} \cap I$, then the $2k+1$ neighbors of a must be in $D \cup P$, so the solution cannot have cost k. Hence, $A_{i,j} \subseteq D \cup P$. Furthermore, the maximum independent set of $\bigcup_{\ell\in\{0,\dots,4\}} B_{i,j,\ell}$ is 2, so $|(\bigcup_{\ell\in\{0,\dots,4\}} B_{i,j,\ell}) \cap (D \cup P)| \geq 8$. Following this reasoning we also observe that if $Q^r_{i,j}$ is not normal, then we have $\text{cost}(Q^r_{i,j}) > 8$. In other words, 8 is a lower bound for the cost of every copy of $Q_{i,j}$, which can only be attained if a copy is normal.

Consider some $j \in \{0, \dots, Fm-1\}$ and $i \in \{1, \dots, n\}$ and suppose that none of the A copies of $Q_{i,j}$ is normal. We will then arrive at a contradiction. Indeed, we have $\text{cost}(\bigcup_r Q^r_{i,j}) \geq 8A + A/2 \geq 8A + 10$. We create another solution by doing the following: take the five vertices $u_{i,5j}, u_{i,5j+1}, \dots, u_{i,5j+4}$, and take in all $Q_{i,j}$ a matching so that $Q_{i,j}$ is normal. This has decreased the total cost, while keeping the solution valid, which should not be possible.

We can therefore assume from now on that for each i, j at least one copy of $Q_{i,j}$ is normal, hence, there exists $\ell \in \{0, \dots, 4\}$ such that $B_{i,j,\ell} \subseteq I$ in that copy.

Recall that $S_{i,j} = \{u_{i,5j}, u_{i,5j+1}, \dots, u_{i,5j+4}\}$. We claim that for all $i \in \{1, \dots, n\}, j \in \{0, \dots, Fm-1\}$, we have $\text{cost}(S_{i,j}) \geq 2$. Indeed, if we consider the normal copy of $Q_{i,j}$ which has $B_{i,j,\ell} \subseteq I$, the two vertices of $B_{i,j,\ell}$ have three neighbors in $S_{i,j}$, and at least one of them must be in D.

In addition, we claim that for all $j \in \{0, \dots, Fm-1\}$ we have $\text{cost}(H_j) \geq 2q(C-1)$. The reasoning here is similar to $Q_{i,j}$, namely, the vertices of W_j cannot belong to I (otherwise we get $2k+1$ vertices in $D \cup P$); and from the $2qC$ vertices in $\bigcup_\sigma Z_{\sigma,j}$ at most $2q$ can belong to I.

We now have the lower bounds we need: $\text{cost}(S_j) \geq 2q(C-1) + 2n + 8An$. ◁

Now, if for some j we have $\text{cost}(S_j) > 2q(C-1) + 2n + 8An$ we will say that j is *problematic*.

▷ Claim 10. There exists a contiguous interval $J \subseteq \{0, \dots, Fm-1\}$ of size at least $m(3n+1)$ in which all $j \in J$ are not problematic.

Proof. Let $L \subseteq \{0, \dots, Fm-1\}$ be the set of problematic indices. We claim that $|L| \leq 2n$. Indeed, we have $\text{cost}(V(G)) = 1 + \sum_{j\in\{0,\dots,Fm-1\}} \text{cost}(S_j) \geq 1 + Fm(2q(C-1) + 2n + 8An) + |L|/2 = k - n + |L|/2$. But since the total cost is at most k, we have $|L|/2 \leq n$.

We will now consider the longest contiguous interval $J \subseteq \{0, \dots, Fm-1\}$ such that all $j \in J$ are not problematic. We have $|J| \geq Fm/(|L|+1) \geq m(3n+1)$. ◁

Before we proceed further, we note that if j is not problematic, then for any $i \in \{1, \dots, n\}$, all edges of M which have an endpoint in $S_{i,j}$, must have their other endpoint also in the main part, that is, they must be edges of the main paths. To see this note that if j is not problematic, all $Q_{i,j}$ are normal, so there are 8 vertices in $A_{i,j} \cap P$ which must be matched to the 8 vertices of $(\bigcup_\ell B_{i,j,\ell}) \cap P$. Similarly, in H_j the $2q(C-1)$ vertices of $W_j \cap P$ must be matched to the $2q(C-1)$ vertices of $(\bigcup_\sigma Z_{\sigma,j}) \cap P$, otherwise we would increase the cost and j would be problematic.

Consider now a non-problematic $j \in J$ and $i \in \{1, \dots, n\}$ such that $\text{cost}(S_{i,j}) = 2$. We claim that the solution must follow one of the five configurations below (see also Figure 1):

(a) $u_{i,5j} \in D$ and $(u_{i,5j+2}, u_{i,5j+3}) \in M$.
(b) $u_{i,5j+1} \in D$ and $(u_{i,5j+3}, u_{i,5j+4}) \in M$.
(c) $u_{i,5j+2} \in D$, $(u_{i,5j+4}, u_{i,5j+5}) \in M$, and $(u_{i,5j-1}, u_{i,5j}) \in M$.
(d) $u_{i,5j+3} \in D$ and $(u_{i,5j}, u_{i,5j+1}) \in M$.
(e) $u_{i,5j+4} \in D$ and $(u_{i,5j+1}, u_{i,5j+2}) \in M$.

Indeed, it is not hard to see that these configurations cover all the cases where exactly one vertex of $S_{i,j}$ is in D and exactly two are in P. This is a condition enforced by the fact that one of the $Q_{i,j}$ copies is normal, and that $\text{cost}(S_{i,j}) = 2$.

▷ Claim 11. There exists a contiguous interval $J' \subseteq \{0, \dots, Fm-1\}$ of size at least m in which all $j \in J'$ are not problematic and for all $j_1, j_2 \in J'$, S_{i,j_1} and S_{i,j_2} are in the same configuration.

Proof. Given the five configurations, we now make the following simple observations, where statements apply for all $i \in \{1, \dots, n\}$ and j such that $j, j+1 \in J$:

- If $S_{i,j}$ is in configuration (a), then $S_{i,j+1}$ is also in configuration (a).
- If $S_{i,j}$ is in configuration (c), then $S_{i,j+1}$ is also in configuration (c).
- If $S_{i,j}$ is in configuration (d), then $S_{i,j+1}$ is in configuration (d) or (a).
- If $S_{i,j}$ is in configuration (b), then $S_{i,j+1}$ is in configuration (b), (d), or (a).

For the first claim, we note that in configuration (a) vertex $u_{i,5j+4}$ is not dominated, forcing the selection of $u_{i,5j+5} \in D$. The second claim is obtained by the observation that all edges of M in this area of the graph must be edges of the path and a parity argument. The third claim is based on the fact that in configuration (d) the edge $(u_{i,5j+4}, u_{i,5j+5})$ must be covered by placing $u_{i,5j+5}$ in $D \cup P$. Finally, configuration (b) cannot be followed by configuration (c), again for parity reasons, nor by configuration (e), because the vertex $u_{i,5j+5}$ would be uncovered.

We will now say for some $i \in \{1, \dots, n\}$, $j \in J$, that j is *shifted* for variable i if $j+1 \in J$ but $S_{i,j}$ and $S_{i,j+1}$ do not have the same configuration. We observe that there cannot exist distinct $j_1, j_2, j_3, j_4 \in J$ such that all of them are shifted for variable i. Indeed, if we draw a directed graph with a vertex for each configuration, and an arc (u, v) expressing the property that the configuration represented by v can follow the one represented by u, if we take into account the observations above, the graph will be a DAG with maximum path length 3. Hence, a configuration cannot shift 4 times, as long as we stay in J (the part of the graph where the minimum local cost is attained everywhere).

By the above, the number of shifted indices $j \in J$ is at most $3n$. Hence, the longest contiguous interval without shifted indices has length at least $|J|/(3n+1) \geq m$. Let J' be this interval. ◁

We are now almost done: we have located an interval $J' \subseteq \{0, \dots, Fm-1\}$ of length at least m where for all $i \in \{1, \dots, n\}$ and all $j_1, j_2 \in J'$ we have the same configuration in S_{i,j_1} and S_{i,j_2}. We now extract an assignment from this in the natural way: if $u_{i,5j+\ell} \in D$, for some $j \in J', \ell \in \{0, \dots, 4\}$, then we set $x_i = \ell$. We claim this satisfies φ. Consider a constraint $c_{j'}$ of φ. There must exist $j \in J'$ such that $j' = j \bmod m$, because $|J'| \geq m$ and J' is contiguous. We therefore check H_j, where there exists σ such that $Z_{\sigma,j} \subseteq I$ (this is because j is not problematic, that is, H_j attains the minimum cost). But because the vertices and incident edges of $Z_{\sigma,j}$ are dominated, it must be the case that the assignment we extracted agrees with σ, hence $c_{j'}$ is satisfied. ◀

We now show that the pathwidth of G is at most $n + O(1)$.

▶ **Lemma 12.** *The pathwidth of G is at most $n + O(q5^q)$.*

Proof. We will show how to build a path decomposition. First, we can add s to all bags, so we focus on the rest of the graph. Second, after removing s from the graph, some vertices become leaves. It is a well-known fact that removing all leaves from a graph can only increase the pathwidth by at most 1. To see this, let G' be the graph obtained after deleting all leaves of G and suppose we have a path decomposition of G' of width w. We obtain a path decomposition of G by doing the following for every leaf v: find a bag of width at most w that contains the neighbor of v and insert after this bag, a copy of the bag with v added. Clearly, the width of the new decomposition is at most $w + 1$. Because of the above we will ignore all vertices of G which become leaves after the removal of s.

For all $j \in \{0, \dots, Fm-1\}$, we will denote $S_j = H_j \cup \bigcup_{i \in \{1,\dots,n\}} \left(S_{i,j} \cup \bigcup_{r \in \{1,\dots,A\}} Q^r_{i,j} \right)$, where $S_{i,j} = \{u_{i,5j}, \dots, u_{i,5j+4}\}$, and $Q^1_{i,j}, \dots, Q^A_{i,j}$ are the A copies of the gadget $Q_{i,j}$. We will show how to build a path decomposition of $G[S_j]$ with the following properties:

- The first bag of the decomposition contains vertices $u_{i,5j}$, for all $i \in \{1, \dots, n\}$.
- The last bag of the decomposition contains vertices $u_{i,5j+4}$, for all $i \in \{1, \dots, n\}$.
- The width of the decomposition is $n + O(q5^q)$.

If we achieve the above then we can obtain a path decomposition of the whole graph: indeed, the sets S_j partition all remaining vertices of the graph, while the only edges not covered by the above decompositions are those between $u_{i,5j+4}$ and $u_{i,5(j+1)}$. We therefore place the decompositions of S_j in order, and then between the last bag of the decomposition of S_j and the first bag of the decomposition of S_{j+1} we have $2n$ "transition" bags, where in each transition step we add a vertex $u_{i,5(j+1)}$ in the bag, and then remove $u_{i,5j+4}$.

Let us now show how to obtain a decomposition of $G[S_j]$, having fixed the contents of the first and last bag. First, H_j has order $O(q5^q)$, so we place all its vertices to all bags. The remaining graph is a union of paths of length 4 with the $Q_{i,j}$ gadgets attached. We therefore have a sequence of $O(n)$ bags, where for each $i \in \{1, \dots, n\}$ we add to the current bag the vertices of $S_{i,j}$, then add and remove one after another whole copies of $Q_{i,j}$, then remove $S_{i,j}$ except for $u_{i,5j+4}$. ◀

We are now ready to present the main result of this section. By putting together Lemmas 7, 8, 12 and the negative result for q-CSP-5 (Lemma 6), we get the following Theorem:

▶ **Theorem 13.** *Under SETH, for all $\varepsilon > 0$, no algorithm solves* Mixed Dominating Set *in time $O^*((5-\varepsilon)^{pw})$, where pw is the input graph's pathwidth.*

Proof. Fix $\varepsilon > 0$ and let q be sufficiently large so that Lemma 6 is true. Consider an instance φ of q-CSP-5. Using our reduction, create an instance (G, k) of Mixed Dominating Set. Thanks to Lemma 7 and Lemma 8, we know that φ is satisfiable if and only if there exists a mixed dominating set of size at most k in G.

Suppose there exists an algorithm which solves Mixed Dominating Set in time $O^*((5-\varepsilon)^{pw})$. With this algorithm and our reduction, we can determine if φ is satisfiable in time $O^*((5-\varepsilon)^{pw})$, where $pw = n+O(q5^q) = n+O(1)$, so the total running time of this procedure is $O^*((5-\varepsilon)^n)$, contradicting the SETH. ◀

4 Exact Algorithm

In this section, we describe an algorithm for the Mixed Dominating Set problem running in time $O^*(1.912^n)$. Let us first give an overview of our algorithm. Consider an instance $G = (V, E)$ of the Mixed Dominating Set problem and fix, for the sake of the analysis, an optimal solution which is a nice mixed dominating set. Such an optimal solution must exist by Lemma 3, so suppose it gives the nice mds partition $V = D \cup P \cup I$.

By Lemma 4, there exists a minimal vertex cover C of G for which $D \subseteq C \subseteq D \cup P$. Our first step is to "guess" C, by enumerating all minimal vertex covers of G. This decreases our search space, since we can now assume that vertices of C only belong in $D \cup P$, and vertices of $V \setminus C$ only belong in $P \cup I$.

For our second step, we branch on the vertices of V, placing them in D, P, or I. The goal of this branching is to arrive at a situation where our partial solution dominates $V \setminus C$. The key idea is that any vertex of C that may belong in D must have at least two private neighbors, hence this allows us to significantly speed up the branching for low-degree vertices of D. Finally, once we have a partial solution that dominates all of $V \setminus C$, we show how to complete this optimally in polynomial time using a maximum matching computation.

We now describe the three steps of our algorithm in order and give the properties we are using step by step. In the remainder we assume that G has no isolated vertices (since these are trivially handled). Therefore, by Lemma 3 there exists an optimal nice mds. Denote the corresponding partition as $V = D \cup P \cup I$.

Step 1. Enumerate all minimal vertex covers of G. For each such vertex cover C we execute the rest of the algorithm. In the end output the best solution found.

Thanks to Lemma 4, there exists a minimal vertex cover C with $D \subseteq C \subseteq D \cup P$. Since we will consider all minimal vertex covers, in the remainder we focus on the case where the set C considered satisfies this property. Let $Z = V \setminus C$. Then Z is an independent set of G. We now get two properties we will use in the branching step of our algorithm:

1. For all $u \in C$, u can be either in D or in P, because $C \subseteq D \cup P$.
2. For all $v \in Z$, v can be either in P or in I, because $D \subseteq C$.

Step 2. Branch on the vertices of V as described below.

The branching step of our algorithm will be a set of Reduction and Branching Rules over the vertices of C or Z. In order to describe a recursive algorithm, it will be convenient to consider a slightly more general version of the problem: in addition to G, we are given three disjoint sets $D_f, P_f, P'_f \subseteq V$, and the question is to build a nice mds partition $V = D \cup P \cup I$ of minimum cost which satisfies the following properties: $D_f \subseteq D \subseteq C$, $P_f \subseteq P \cap C$, and $P'_f \subseteq P \cap Z$. Clearly, if $D_f = P_f = P'_f = \emptyset$ we have the original problem and all properties are satisfied. We will say that a branch where all properties are satisfied is *good*, and our proof of correctness will rely on the fact that when we branch on a good instance, at least one of the produced branches is good. The intuitive meaning of these sets is that when we decide in a branch that a vertex belongs in D or in P in the optimal partition we place it respectively in D_f, P_f or P'_f (depending on whether the vertex belongs in C or Z).

We now describe a series of Rules which, given an instance of Mixed Dominating Set and three sets D_f, P_f, P'_f, will recursively produce subinstances where vertices are gradually placed into these sets. Our algorithm will consider the Reduction and Branching Rules in order and apply the first Rule that can be applied. Note that we say that a vertex u is *decided* if it is in one of the sets $D_f \subseteq D$, $P_f \subseteq P$, or $P'_f \subseteq P$. All the other vertices are considered *undecided*.

Throughout the description that follows, we will use U to denote the set of undecided vertices which are not dominated by D_f, that is, $U := V \setminus (D_f \cup P_f \cup P'_f \cup (N(D_f) \cap Z))$. We will show that when no rule can be applied, U is empty, that is, all vertices are decided or dominated by D_f. In the third step of our algorithm we will show how to complete the solution in polynomial time when U is empty. Since our Rules do not modify the graph, we will describe the subinstances we branch on by specifying the tuple (D_f, P_f, P'_f).

To ease notation, let $U_C = U \cap C$ and $U_Z = U \cap Z$. Recall that for $u \in V$, we use $d_{U_C}(u)$ and $d_{U_Z}(u)$ to denote the size of the sets $N(u) \cap U_C = N_{U_C}(u)$ and $N(u) \cap U_Z = N_{U_Z}(u)$, respectively.

Reduction Rule (R1): If there exists $u \in U_C$ such that $d_{U_Z}(u) \leq 1$, then put u in P_f, that is, recurse on the instance $(D_f, P_f \cup \{u\}, P'_f)$.

Reduction Rule (R2): If there exists $v \in U_Z$ such that $d_{U_C}(v) = 0$, then put u in P'_f, that is, recurse on the instance $(D_f, P_f, P'_f \cup \{v\})$.

Branching Rule (B1): If there exists $u \in U_C$ such that $d_{U_Z}(u) \geq 4$, then branch on the following two subinstances: $(D_f \cup \{u\}, P_f, P'_f)$ and $(D_f, P_f \cup \{u\}, P'_f)$.

Note that we may now assume that all vertices of U_C have $d_{U_Z} \in \{2, 3\}$. The following two rules eliminate vertices $u \in U_C$ with $d_{U_Z}(u) = 2$.

Branching Rule (B2.1): If there exists $u_1, u_2 \in U_C$ such that $d_{U_Z}(u_1) = 3$, $d_{U_Z}(u_2) = 2$, and $N_{U_Z}(u_1) \cap N_{U_Z}(u_2) \neq \emptyset$ then branch on the following instances: $(D_f \cup \{u_1\}, P_f \cup \{u_2\}, P'_f)$ and $(D_f, P_f \cup \{u_1\}, P'_f)$.

Branching Rule (B2.2): If there exists $u \in U_C$ with $d_{U_Z}(u) = 2$ we branch on the instances $(D_f \cup \{u\}, P_f, P'_f)$ and $(D_f, P_f \cup \{u\}, P'_f)$.

We now have that all vertices $u \in U_C$ have $d_{U_Z}(u) = 3$. Let us now branch on vertices of U_Z to ensure that these also do not have too low degree.

Branching Rule (B3.1): If there exists $v \in U_Z$ with $d_{U_C}(v) = 1$ let $N_{U_C}(v) = \{u\}$. We branch on the instances $(D_f \cup \{u\}, P_f, P'_f)$ and $(D_f, P_f \cup \{u\}, P'_f)$.

Branching Rule (B3.2): If there exists $v \in U_Z$ with $d_{U_C}(v) = 2$ let $N_{U_C}(v) = \{u_1, u_2\}$. We branch on the instances $(D_f \cup \{u_1\}, P_f, P'_f)$, $(D_f \cup \{u_2\}, P_f \cup \{u_1\}, P'_f)$, and $(D_f, P_f \cup \{u_1, u_2\}, P'_f)$.

If we cannot apply any of the above Rules, for all $u \in U_C$ we have $d_{U_Z}(u) = 3$ and for all $v \in U_Z$ we have $d_{U_C}(v) \geq 3$. We now consider three remaining cases: (i) there exists a C_4 made up of two vertices of U_C and two vertices of U_Z (ii) there exists a vertex $v \in U_Z$ with $d_{U_C}(v) = 3$ (iii) everything else.

Branching Rule (B4): If there exist $u_1, u_2 \in U_C$ and $v_1, v_2 \in U_Z$ with $(u_i, v_j) \in E$ for all $i, j \in \{1, 2\}$, then we branch on the instances $(D_f \cup \{u_1\}, P_f \cup \{u_2\}, P'_f)$ and $(D_f, P_f \cup \{u_1\}, P'_f)$.

Branching Rule (B5): If there exists $v \in U_Z$ with $d_{U_C}(v) = 3$, let $N_{U_C}(v) = \{u_1, u_2, u_3\}$ and for $i \in \{1, 2, 3\}$ let $X_i = \{w \in U_C \setminus \{u_1, u_2, u_3\} \mid N(w) \cap N(u_i) \cap (U_Z \setminus \{v\}) \neq \emptyset\}$, that is, X_i is the set of vertices of U_C that share a neighbor with u_i in U_Z other than v. Then we branch on the following 8 instances: (i) the instance $(D_f, P_f \cup \{u_1, u_2, u_3\}, P'_f \cup \{v\}\}$

(ii) for $i \in \{1,2,3\}$, we produce the instances $(D_f \cup \{u_i\}, P_f \cup (\{u_1,u_2,u_3\} \setminus \{u_i\}), P'_f)$ (iii) for $i,j \in \{1,2,3\}$, with $i < j$ we produce the instances $(D_f \cup \{u_i,u_j\}, P_f \cup (\{u_1,u_2,u_3\} \setminus \{u_i,u_j\}) \cup X_i \cup X_j, P'_f)$ (iv) we produce the instance $(D_f \cup \{u_1,u_2,u_3\}, P_f \cup X_1 \cup X_2 \cup X_3, P'_f)$.

Branching Rule (B6): Consider $u \in U_C$ and let $N_{U_Z}(u) = \{v_1,v_2,v_3\}$. We branch on the following instances: $(D_f, P_f \cup \{u\}, P'_f)$, $(D_f \cup \{u\}, P_f \cup N_{U_1}(v_1) \setminus \{u\}, P'_f)$, and $(D_f \cup \{u\}, P_f \cup (N_{U_1}(v_2) \cup N_{U_1}(v_3)) \setminus \{u\}, P'_f)$.

Our algorithm applies the above Rules in order as long as possible. Before proceeding to explain what happens when no Rule is applicable, let us first establish two useful correctness properties. We will say that a tuple (D_f, P_f, P'_f) is *good* if $D_f \subseteq D$, $P_f \subseteq P \cap C$, and $P'_f \subseteq P \setminus C$.

▶ **Lemma 14.** *If we apply the first Rule that can be applied on an instance characterized by a good tuple, then we produce at least one instance characterized by a good tuple.*

Proof. We consider the Rules in order. For Reduction Rule 1, observe that all neighbors of u in U_1 cannot be private neighbors of u since $U_C \subseteq C \subseteq D \cup P$, and because $d_{U_Z}(u) \leq 1$, the vertex u can have at most one private neighbor, so it must be the case that $u \in P$. For Reduction Rule 2, v must be dominated, but it has no neighbor in U_C, so it must be the case that $v \in P$. Branching Rule B1 is trivially correct from $C \subseteq D \cup P$.

Branching Rule B2.1 is correct because if $u_1 \in D$, then u_2 cannot have two private neighbors and it is forced to be in P. Branching Rule B2.2 is trivially correct again from $C \subseteq D \cup P$.

Again, from $C \subseteq D \cup P$, Branching Rule B3.1 is trivially correct. Branching rule B3.2 is correct since we have the three following cases: $u_1 \in D$; or $u_1 \in P$ and $u_2 \in D$; or u_1 and $u_2 \in P$.

Branching Rule B4 is correct because if $u_1 \in D$, then u_2 cannot have two private neighbors since $d_{U_Z}(v) = 3$.

Branching Rule B5 is correct since we have the following cases: all vertices u_1, u_2 and u_3 are in P ; or exactly one of them is in D ; or exactly two of them are in D ; or all of them are in D. Note first that u_1, u_2 and u_3 only share v as neighbor in U_Z since Branching Rule B4 is not triggered. In the first case, v has to be dominated so it must be the case that $v \in P$. In the second case, the two vertices not in D necessarily are in P. In the third case, since u_i and u_j share v as common neighbor and both have exactly three neighbors in U_Z, the vertices of X_i and X_j have to be in P because otherwise u_i and u_j do not have two private neighbors. In the last case, and for the same reason, the vertices of X_1, X_2, and X_3 have to be in P.

Finally, Branching Rule B6 is correct because if $u \in D$, then either v_1 is one of its private neighbors, or both v_2 and v_3 are its private neighbors. ◀

▶ **Lemma 15.** *If none of the Rules can be applied then $U = \emptyset$.*

Proof. Observe that by applying rules R1, B1, B2.2, B6, we eventually eliminate all vertices of U_C, since these rules alone cover all the cases for $d_{U_Z}(u)$ for any $u \in U_C$. So, if none of these rules applies, U_C is empty. But then applying R2 will also eliminate U_Z, which makes all of U empty. ◀

Step 3. When U is empty, reduce the problem to Maximum Matching.

We now show how to complete the solution in polynomial time.

▶ **Lemma 16.** *Let (D_f, P_f, P'_f) be a good tuple such that no Rule can be applied. Then it is possible to construct in polynomial time a mixed dominating set of size $|D| + \frac{|P|}{2}$.*

Proof. Because no Rule can be applied, by Lemma 15 we have $U = V \setminus (D_f \cup P_f \cup P'_f \cup (N(D_f) \setminus C)) = \emptyset$.

Let M be a maximum matching of $G[P_f \cup P'_f]$. Then, we claim that $|D| + |P|/2 \geq |D_f| + |P_f \cup P'_f| - |M|$. First, $|D| \geq |D_f|$ because $D_f \subseteq D$. We now claim that $P \setminus (P_f \cup P'_f)$ is an independent set. Indeed, $P \setminus (P_f \cup P'_f)$ is a set of undecided vertices, and because U is empty, the only undecided vertices are those in $Z \cap N(D_f)$, which is an independent set. Consider now a perfect matching M' of $G[P]$, and let M'' be the set of edges of that matching that have both endpoints in $P_f \cup P'_f$. Clearly, $|M''| \leq |M|$. Let $P' = (P_f \cup P'_f) \setminus V(M'')$. By the definitions of M', M'', P', and the fact that $P \setminus (P_f \cup P'_f)$ is an independent set, we have: $|P|/2 = |M'| = |M''| + |P'|$. By this, we get: $|P|/2 = |M''| + |P_f \cup P'_f| - |V(M'')| = |P_f \cup P'_f| - |M''| \geq |P_f \cup P'_f| - |M|$. By summing this last inequality with $|D| \geq |D_f|$, we get: $|D| + |P|/2 \geq |D_f| + |P_f \cup P'_f| - |M|$.

We will now show how to construct a valid mixed dominating set of size $|D_f| + |P_f \cup P'_f| - |M|$ in polynomial time, where again M is a maximum matching of $G[P_f \cup P'_f]$. Specifically, we select all vertices of D_f, all edges of M, and an edge incident on each unmatched vertex of $P_f \cup P'_f$. The size of such a solution is $|D_f| + |M| + (|P_f \cup P'_f| - 2|M|)$, which is equal to the bound we promised.

To conclude, let us explain why the solution we have produced is a valid mixed dominating set (even though it is not necessarily a nice mds). First, the solution we produced puts all vertices of C in $D \cup P$, therefore, since C is a vertex cover, all edges are covered. Second, since all Rules were exhaustively applied, our tuple gives $U = \emptyset$, which implies that all vertices of Z are either in $N(D_f)$ or in P'_f, therefore dominated. ◀

We give a small overview of the analysis of the running time of our exact algorithm. First, enumerating all minimal vertex covers takes times at most $O^*(3^{n/3})$, which is also an upper bound on the number of such covers [22]. Moreover, we observe that we can decide if a Rule applies in polynomial time, and the algorithm of Lemma 16 runs in polynomial time. We therefore only need to bound the number of subinstances the branching step will produce, as a function of n.

We define our measure of progress as the size of the set $\{u \in U_C \mid d_{U_Z}(u) \geq 2\} \cup \{v \in U_Z \mid d_{U_C}(v) \geq 1\}$. In other words, we count the undecided vertices of U_C that have at least two undecided, non-dominated vertices in Z, and the undecided, non-dominated vertices of Z that have at least one undecided neighbor in C. This is motivated by the fact that undecided vertices that do not respect these degree bounds are eliminated by the Reduction Rules and hence do no affect the running time. Let l denote the number of vertices that we counted according to this measure. Clearly, we have $l \leq n$.

Of all the above rules, the worst case is given by Branching Rule B5, which leads to a complexity of 1.3252^l. Taking into account the cost of enumerating all minimal vertex covers and the fact that $l \leq n$, the running time of our algorithm is $O^*(3^{n/3} \cdot 1.3252^n) = O^*(1.912^n)$.

▶ **Theorem 17.** *Mixed Dominating Set can be solved in time $O^*(1.912^n)$ and polynomial space.*

5 FPT Algorithm

In this section we describe an algorithm for k-Mixed Dominating Set running in time $O^*(3.510^k)$, where k is the value of the optimal solution. Our algorithm is based on a branching procedure very similar to the one used in [26], which runs in time $O^*(4.172^k)$. We only sketch the different branching rules and explain, on a high level, how the notion of nice dominating sets allows us to obtain the improved running time.

We fix for the analysis an optimal nice mds and its partition $V = D \cup P \cup I$. The branching algorithm will gradually build two sets D_f, P_f which are the vertices decided to be in D, P respectively. Let $U = V \setminus (D_f \cup P_f)$ be the set of undecided vertices. As noted in [26], a basic branching algorithm considers for each $u \in U$ the cases $u \in D_f$, $u \in P_f$, and all partitions of $N_U(u)$ into D_f, P_f. This leads to a performance similar to that of [12] ($O^*(7.465^k)$). The key idea of [26] is to identify the importance of the set $U^* = U \setminus N(D_f)$ of undecided, undominated vertices. Branching on U^* is faster because we no longer need to consider the case $N(u) \subseteq P_f$. Once $U^* = \emptyset$, the problem becomes much easier.

Our improvement is based on the fact that (by Lemma 3) each $u \in D$ has two private neighbors. This speeds up branching on U^*, as we have: (i) if $d_{U^*}(u) < 2$, then the branch $u \in D_f$ need not be considered (ii) if $d_{U^*}(u) = 2$ then in the branch where $u \in D_f$ we may assume that the two vertices $v_1, v_2 \in N(u) \cap U^*$ are private neighbors of u, so their undecided neighbors are automatically placed in P_f (iii) if $d_{U^*}(u) > 3$, for the branch where $u \in D_f$ we can consider sub-branches where we decide which are the private neighbors of u, placing the neighbors of these vertices in P_f. A key element of our analysis is that, because we have sped up the branching on low-degree ($d_{U^*}(u) \leq 2$) vertices, in subsequent branches we are allowed to assume that all neighbors of u have several undecided neighbors, increasing the profit of guessing that v is a private neighbor of u. Using these ideas we speed up the branching on U^* and the remainder of the algorithm follows along similar lines to [26].

▶ **Theorem 18.** *k-Mixed Dominating Set can be solved in time $O^*(3.510^k)$.*

References

1 Yousef Alavi, M. Behzad, Linda M. Lesniak-Foster, and E. A. Nordhaus. Total matchings and total coverings of graphs. *Journal of Graph Theory*, 1(2):135–140, 1977.

2 Yousef Alavi, Jiuqiang Liu, Jianfang Wang, and Zhongfu Zhang. On total covers of graphs. *Discrete Mathematics*, 100(1-3):229–233, 1992.

3 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In *IPEC*, volume 63 of *LIPIcs*, pages 8:1–8:23, 2016.

4 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015.

5 Szymon Dudycz, Mateusz Lewandowski, and Jan Marcinkowski. Tight approximation ratio for minimum maximal matching. In *IPCO*, volume 11480 of *LNCS*. Springer, 2019.

6 Paul Erdös and Amram Meir. On total matching numbers and total covering numbers of complementary graphs. *Discrete Mathematics*, 19(3):229–233, 1977.

7 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Intractability of clique-width parameterizations. *SIAM J. Comput.*, 39(5):1941–1956, 2010.

8 Tesshu Hanaka, Ioannis Katsikarelis, Michael Lampis, Yota Otachi, and Florian Sikora. Parameterized orientable deletion. In *SWAT*, volume 101 of *LIPIcs*, pages 24:1–24:13, 2018.

9 Pooya Hatami. An approximation algorithm for the total covering problem. *Discussiones Mathematicae Graph Theory*, 27(3):553–558, 2007.

10 Teresa W. Haynes, Stephen T. Hedetniemi, and Peter J. Slater. *Fundamentals of domination in graphs*, volume 208 of *Pure and applied mathematics*. Dekker, 1998.

11 Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *CIAC*, volume 10236 of *LNCS*, pages 345–356, 2017.

12 Pallavi Jain, M. Jayakrishnan, Fahad Panolan, and Abhishek Sahu. Mixed dominating set: A parameterized perspective. In *WG*, volume 10520 of *LNCS*, pages 330–343. Springer, 2017.

13 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structurally parameterized d-scattered set. In *WG*, volume 11159 of *LNCS*, pages 292–305. Springer, 2018.

14 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discr. Applied Math.*, 264:90–117, 2019.

15 Michael Lampis. Finer tight bounds for coloring on clique-width. In *ICALP*, volume 107 of *LIPIcs*, pages 86:1–86:14, 2018.

16 James K. Lan and Gerard Jennhwa Chang. On the mixed domination problem in graphs. *TCS*, 476:84–93, 2013.

17 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018.

18 Jayakrishnan Madathil, Fahad Panolan, Abhishek Sahu, and Saket Saurabh. On the complexity of mixed dominating set. In *CSR*, volume 11532 of *LNCS*, pages 262–274. Springer, 2019.

19 Aniket Majumdar. Neighborhood hypergraphs: A framework for covering and packing parameters in graphs. *PhD thesis, Clemson University*, 1993.

20 David Manlove. On the algorithmic complexity of twelve covering and independence parameters of graphs. *Discrete Applied Mathematics*, 91(1-3):155–175, 1999.

21 Amram Meir. On total covering and matching of graphs. *JCTS B*, 24(2):164–168, 1978.

22 John W Moon and Leo Moser. On cliques in graphs. *Israel j. of Math.*, 3(1):23–28, 1965.

23 Uri N. Peled and Feng Sun. Total matchings and total coverings of threshold graphs. *Discrete Applied Mathematics*, 49(1-3):325–330, 1994.

24 M. Rajaati, Mohammad Reza Hooshmandasl, Michael J. Dinneen, and Ali Shakiba. On fixed-parameter tractability of the mixed domination problem for graphs with bounded tree-width. *Discrete Mathematics & Theoretical Computer Science*, 20(2), 2018.

25 M. Rajaati, P. Sharifani, Ali Shakiba, Mohammad Reza Hooshmandasl, and Michael J. Dinneen. An efficient algorithm for mixed domination on generalized series-parallel graphs. *CoRR*, abs/1708.00240, 2017. `arXiv:1708.00240`.

26 Mingyu Xiao and Zimo Sheng. Improved parameterized algorithms for mixed domination. In *AAIM*, volume 11640 of *LNCS*, pages 304–315. Springer, 2019.

27 Yancai Zhao, Liying Kang, and Moo Young Sohn. The algorithmic complexity of mixed domination in graphs. *Theor. Comput. Sci.*, 412(22):2387–2392, 2011.