Abstract. The linear maximum induced matching width (LMIM-width) of a graph is a width parameter based on the maximum induced matching in some of its subgraphs. In this paper we study output-polynomial enumeration algorithms on graphs of bounded LMIM-width and graphs of bounded local LMIM-width. In particular, we show that all 1-minimal \((\sigma, \rho)\)-dominating sets, and hence all minimal dominating sets, of graphs of bounded LMIM-width can be enumerated with polynomial (linear) delay using polynomial space. Furthermore, we show that all minimal dominating sets of a unit square graph can be enumerated in incremental polynomial time.

1 Introduction

Enumeration is at the heart of computer science and combinatorics. Enumeration algorithms for graphs and hypergraphs typically deal with listing all vertex subsets or edge subsets satisfying a given property. As the size of the output is often exponential in the size of the input, it is customary to measure the running time of enumeration algorithms in the size of the input plus the size of the output. If the running time of an algorithm is bounded by a polynomial in the size of the input plus the size of the output, then the algorithm is called output-polynomial. A large number of such algorithms have been given over the last 30 years; many of them solving problems on graphs and hypergraphs [7,8,9,12,18,19,20,22]. It is also possible to show that certain enumeration problems have no output-polynomial time algorithm unless \(P = NP\) [18,19,20].

Recently Kanté et al. showed that the famous longstanding open question whether there is an output-polynomial algorithm to enumerate all minimal transversals of a hypergraph is equivalent to the question whether there is an output-polynomial algorithm to enumerate all minimal dominating sets of a graph [13]. Although the main question remains open, a large number of results have been
obtained on graph classes. Output-polynomial algorithms to enumerate all minimal dominating sets exist for graphs of bounded treewidth and of bounded clique-width [6], interval graphs [7], strongly chordal graphs [7], planar graphs [9], degenerate graphs [9], split graphs [13], path graphs [14], permutation graphs [15], line graphs [10,14,17], chordal bipartite graphs [11], chordal graphs [16] and graphs of girth at least 7 [10].

In this paper, we extend the above results to graphs of bounded linear maximum induced matching width (LMIM-width), which is a linearized version of the notion of maximum induced matching width introduced by Vatshelle [24]. Belmonte and Vatshelle showed that several important graph classes, among them interval, circular-arc and permutation graphs, have bounded LMIM-width [1]. Polynomial-time algorithms solving optimization problems on such graph classes have been studied in [4,24].

In this paper, we study two ways of using bounded LMIM-width in enumeration algorithms. In Section 3 we study the enumeration problem corresponding to an extended and colored version of the well-known $p_\sigma,\rho,q$-domination problem, asking to enumerate all 1-minimal $p_\sigma,\rho,q$-dominating sets. This includes the enumeration of all minimal (total) dominating sets on graphs of bounded LMIM-width. We establish as our main result an enumeration algorithm with polynomial (linear) delay and polynomial space for this problem. Our algorithm uses the enumeration (and counting) of paths in directed acyclic graphs. In Section 4 we study the enumeration of all minimal dominating sets in unit square graphs. We first show that any $r$-neighborhood in such graphs have LMIM-width bounded by $O(r^2)$. Then we show how to adapt the so-called flipping method developed by Golovach et al. [10] to enumerate all minimal dominating sets of a unit square graph in incremental polynomial time.

2 Definitions and preliminaries

**Graphs.** The power set of a set $V$ is denoted by $2^V$. For two sets $A$ and $B$ we let $A\setminus B$ be the set $\{x \in A \mid x \notin B\}$, and if $X$ is a subset of a ground set $V$, we let $\overline{X}$ be the set $V \setminus X$. We often write $x$ to denote the singleton set $\{x\}$. We denote by $\mathbb{N}$ the set of positive or null integers, and let $\mathbb{N}^*$ be $\mathbb{N}\setminus\{0\}$.

A graph $G$ is a pair $(V(G), E(G))$ with $V(G)$ its set of vertices and $E(G)$ its set of edges. An edge between two vertices $x$ and $y$ is denoted by $xy$ (respectively $yx$). The subgraph of $G$ induced by a subset $X$ of its vertex set is denoted by $G[X]$. The set of vertices that is adjacent to $x$ is denoted by $N_G(x)$, and we let $N_G[x]$ be the set $N_G(x) \cup \{x\}$. For $U \subseteq V(G)$, $N_G[U] = \bigcup_{v \in U} N_G[v]$ and $N_G(U) = N_G[U] \setminus U$. For a vertex $x$ and a positive integer $r$, $N_G[r](x)$ denotes the set of vertices at distance at most $r$ from $x$. For two disjoint subsets $A$ and $B$ of $V(G)$, let $G[A, B]$ denote the graph with vertex set $A \cup B$ and edge set $\{uv \in E(G) \mid u \in A, v \in B\}$. Clearly, $G[A, B]$ is a bipartite graph and $\{A, B\}$ is its bipartition. Recall that a set of edges $M$ is an induced matching if the end-vertices of distinct edges of $M$ are different and not adjacent. We denote by $\text{mim}_G(A, B)$ the size of a maximum induced matching in $G[A, B]$. 
Let $G$ be a graph, and let $\text{Red}, \text{Blue} \subseteq V(G)$ such that $\text{Red} \cup \text{Blue} = V(G)$. We refer to the vertices of $\text{Red}$ as the red vertices, the vertices of $\text{Blue}$ as the blue vertices, and we say that $G$ together with given sets $\text{Red}$ and $\text{Blue}$ is a colored graph. For simplicity, whenever we say that $G$ is a colored graph, it is assumed that the sets $\text{Red}$ and $\text{Blue}$ are given. Notice that $\text{Red}$ and $\text{Blue}$ are not necessarily disjoint. In particular, it can happen that $\text{Red} = \text{Blue} = V(G)$; a non-colored graph $G$ can be seen as a colored graph with $\text{Red} = \text{Blue} = V(G)$.

A graph $G$ is an (axis-parallel) unit square graph if it is an intersection graph of squares in the plane with their sides parallel to the coordinate axis. These graphs also are known as the graphs of cubicity 2. We use the following equivalent definition, see e.g. [5], in which each vertex $v$ of $G$ is represented by a point in $\mathbb{R}^2$. A graph $G$ is a unit square graph if there is a function $f: V(G) \to \mathbb{R}^2$ such that two vertices $u, v \in V(G)$ are adjacent in $G$ if and only if $\|f(u) - f(v)\|_\infty < 1$, where the norm $\|\cdot\|_\infty$ is the $L_\infty$ norm. For a vertex $v \in V(G)$, we let $x_f(v)$ and $y_f(v)$ denote the $x$ and $y$-coordinate of $f(v)$ respectively. We say that the point $(x_f(v), y_f(v))$ represents $v$. The function $f$ is called a realization of the unit square graph. It is straightforward to see that for any unit square graph $G$, there is a realization $f: V(G) \to \mathbb{Q}^2$. We always assume that a unit square graph is given with its realization. Indeed, it is NP-hard to recognize unit square graphs [3]. We refer to the survey of Brandstädt, Le and Spinrad [2] for the definitions of all other graph classes mentioned in our paper.

**Enumeration.** Let $\mathcal{D}$ be a family of subsets of the vertex set of a given graph $G$ on $n$ vertices and $m$ edges. An enumeration algorithm for $\mathcal{D}$ lists the elements of $\mathcal{D}$ without repetitions. The running time of an enumeration algorithm $\mathcal{A}$ is said to be output polynomial if there is a polynomial $p(x, y)$ such that all the elements of $\mathcal{D}$ are listed in time bounded by $p((n + m), |\mathcal{D}|)$. Assume now that $D_1, \ldots, D_\ell$ are the elements of $\mathcal{D}$ enumerated in the order in which they are generated by $\mathcal{A}$. Let us denote by $T(\mathcal{A}, i)$ the time $\mathcal{A}$ requires until it outputs $D_i$, also $T(\mathcal{A}, \ell + 1)$ is the time required by $\mathcal{A}$ until it stops. Let $\text{delay}(\mathcal{A}, 1) = T(\mathcal{A}, 1)$ and $\text{delay}(\mathcal{A}, i) = T(\mathcal{A}, i) - T(\mathcal{A}, i - 1)$. The delay of $\mathcal{A}$ is $\max\{\text{delay}(\mathcal{A}, i)\}$. Algorithm $\mathcal{A}$ runs in incremental polynomial time if there is a polynomial $p(x, i)$ such that $\text{delay}(\mathcal{A}, i) \leq p(n + m, i)$. Furthermore $\mathcal{A}$ is a polynomial delay algorithm if there is a polynomial $p(x)$ such that the delay of $\mathcal{A}$ is at most $p(n + m)$. Finally $\mathcal{A}$ is a linear delay algorithm if $\text{delay}(\mathcal{A}, 1)$ is bounded by a polynomial in $n + m$ and $\text{delay}(\mathcal{A}, i)$ is bounded by a linear function in $n + m$.

**Linear induced matching width.** The notion of the maximum induced matching width was introduced by Vatshelle [24] (see also [1]). We will give the definition in terms of colored graphs and restrict ourselves to the case of linear maximum induced matching width. Let $G$ be a colored $n$-vertex graph with $n \geq 2$ and let $x_1, \ldots, x_n$ be a linear ordering of its vertex set. For each $1 \leq i \leq n$, we let $A_i = \{x_1, x_2, \ldots, x_i\}$ and $\bar{A}_i = \{x_{i+1}, x_{i+2}, \ldots, x_n\}$. The maximum induced matching width (MIM-width) of $x_1, \ldots, x_n$ is

$$\max\{\max\{\text{mim}_G(A_i \cap \text{Red}, \bar{A}_i \cap \text{Blue}), \text{mim}_G(A_i \cap \text{Blue}, \bar{A}_i \cap \text{Red})\} \mid 1 \leq i \leq n\}.$$
The linear maximum induced matching width (LMIM-width) of G, denoted by \( \text{lmimw}(G) \), is the minimum value of the MIM-width taken over all linear orderings of G.

Belmonte and Vatshelle [1] proved that several important graph classes have bounded linear maximum induced matching width. For example, the LMIM-width of an interval graph is 1 and the LMIM-width of a permutation graph is at most 2.

\((\sigma, \rho)\)-domination. The \((\sigma, \rho)\)-dominating set notion was introduced by Telle and Proskurowski [23] as a generalization of dominating sets. Indeed, many NP-hard domination type problems such as the problems \(d\)-Dominating Set, Independent Dominating Set and Total Dominating Set are special cases of the \((\sigma, \rho)\)-Dominating Set Problem. See [4, Table 1] for more examples. For technical reasons, we introduce \textbf{Red} \((\sigma, \rho)\)-domination. Let \( \sigma \) and \( \rho \) be finite or co-finite subsets of \( \mathbb{N} \). We say that a set \( D \subseteq V(G) \) \((\sigma, \rho)\)-dominates \( U \subseteq V(G) \) if it \((\sigma, \rho)\)-dominates every \( u \in U \), i.e., for each \( u \in U \), \(|N_G(u) \cap D| \in \sigma \) if \( u \in D \), otherwise \(|N_G(u) \cap D| \in \rho \).

Let \( G \) be a colored graph. A set of vertices \( D \subseteq \text{Red} \) is a \textbf{Red} \((\sigma, \rho)\)-dominating set if \( D \) \((\sigma, \rho)\)-dominates \textbf{Blue}. If \( \text{Red} = \text{Blue} = V(G) \), then a \textbf{Red} \((\sigma, \rho)\)-dominating set is a \((\sigma, \rho)\)-dominating set.

Notice that if \( \sigma = \mathbb{N} \) and \( \rho = \mathbb{N}^\ast \), then a set \( D \subseteq V(G) \) \((\sigma, \rho)\)-dominates a vertex \( u \) if \( u \in D \) or \( u \) is adjacent to a vertex of \( D \), i.e., the notion of \((\sigma, \rho)\)-domination coincides with the classical domination in this case. Whenever we consider this case, we simply write that a set \( D \) dominates a vertex or set and \( D \) is a \((\text{Red})\) dominating set omitting \((\sigma, \rho)\). We are interested in \textbf{Red} dominating sets because that is what we actually need in Section 4.

A \textbf{Red} \((\sigma, \rho)\)-dominating set \( D \) of a graph \( G \) is said \textit{minimal} if for any proper subset \( D' \subset D \), \( D' \) is not a \textbf{Red} \((\sigma, \rho)\)-dominating set, and we say that \( D \) is \textit{1-minimal} if for each vertex \( x \in D \), \( D \setminus \{x\} \) is not a \textbf{Red} \((\sigma, \rho)\)-dominating set. Clearly, every minimal \textbf{Red} \((\sigma, \rho)\)-dominating set is 1-minimal, but the converse is not true for arbitrary \( \sigma \) and \( \rho \).

Because our aim is to enumerate 1-minimal \textbf{Red} \((\sigma, \rho)\)-dominating sets, we need some certificate that a considered set is 1-minimal. Let \( D \) be a \textbf{Red} \((\sigma, \rho)\)-dominating set of a colored graph \( G \). For a vertex \( u \in D \), the vertex \( v \in \text{Blue} \) is its \textit{certifying vertex} (or a \textit{certificate}) if \( v \) is not \((\sigma, \rho)\)-dominated by \( D \setminus \{u\} \).

Notice that because \( D \) is a \textbf{Red} \((\sigma, \rho)\)-dominating set, if \( v \) is a certificate for \( u \), then \( v \in N_G[u] \). Observe also that, for some pairs \((\sigma, \rho)\), a vertex may be a certificate for many vertices and it can be a certificate for itself. Notice that in the case of the classical domination, certificates are usually called \textit{privates} because they are certificates for exactly one vertex, including itself. It is straightforward to show the following.

**Lemma 1.** A set \( D \subseteq \text{Red} \) is a 1-minimal \textbf{Red} \((\sigma, \rho)\)-dominating set of a colored graph \( G \) if and only if each vertex \( u \in D \) has a certificate.

**Lemma 2.** Let \( D \) be a \textbf{Red} \((\sigma, \rho)\)-dominating set of \( G \). If \( v \) is a certificate for \( u \in D \), then \( v = u \) or \( v \) is a certificate for all vertices of \( N_G(v) \cap D \).

4
3 Enumerations for graphs of bounded LMIM-width

In this section we prove the following, which generalizes the results in [15].

**Theorem 1.** Let \((\sigma, \rho)\) be a pair of finite or co-finite subsets of \(\mathbb{N}\) and let \(c\) be a positive integer. For a colored graph \(G\) given with a linear ordering of \(V(G)\) of MIM-width at most \(c\), one can count in time bounded by \(O(n^c)\), and enumerate with linear delay, all 1-minimal Red \((\sigma, \rho)\)-dominating sets of \(G\).

**Corollary 1.** Let \((\sigma, \rho)\) be a pair of finite or co-finite subsets of \(\mathbb{N}\). Then, for every colored graph \(G\) in one of the following graph classes, we can count in polynomial time, and enumerate with linear delay all 1-minimal Red \((\sigma, \rho)\)-dominating sets of \(G\): interval graphs, permutation graphs, circular-arc graphs, circular permutation graphs, trapezoid graphs, convex graphs, and for fixed \(k\), \(k\)-polygon graphs, Dilworth-\(k\) graphs and complements of \(k\)-degenerate graphs.

The following corollary improves some known results in the enumeration of minimal transversals of interval and circular-arc hypergraphs where only an incremental polynomial time algorithm was known (see e.g. [21]).

**Corollary 2.** For every hypergraph \(H\) being an interval hypergraph or a circular-arc hypergraph one can count in polynomial time, and enumerate with linear delay, all minimal transversals of \(H\).

The remaining part of the section is devoted to the main ideas of the proof of Theorem 1. Throughout this section let \((\sigma, \rho)\) be a fixed pair of finite or co-finite subsets of \(\mathbb{N}\) and let \(G\) be a fixed \(n\)-vertex colored graph with \(n \geq 2\). Let \(x_1, \ldots, x_n\) be a fixed linear ordering of the vertex set of \(G\) such that the maximum induced matching width of \(x_1, \ldots, x_n\) is bounded by a constant \(c\). Furthermore, for all \(i \in \{1, 2, \ldots, n\}\), we let \(A_i = \{x_1, x_2, \ldots, x_i\}\) and \(\overline{A}_i = \{x_{i+1}, x_{i+2}, \ldots, x_n\}\).

Let \(d(\mathbb{N}) = 0\). For every finite set \(\mu \subseteq \mathbb{N}\), let \(d(\mu) = 1 + \max\{a | a \in \mu\}\), and for every co-finite set \(\mu \subseteq \mathbb{N}\), let \(d(\mu) = 1 + \max\{a | a \in \mathbb{N}\setminus\mu\}\). For finite or co-finite subsets \(\sigma\) and \(\rho\) of \(\mathbb{N}\), we let \(d(\sigma, \rho) = \max(d(\sigma), d(\rho))\). As pointed out in [4] given a subset \(D\) of Red, we can check if \(D\) is a Red \((\sigma, \rho)\)-dominating set by computing \(|D \cap N_G(x)|\) up to \(d(\sigma, \rho)\) for each vertex \(x\) in Blue. We define \(\sigma^* = \sigma \cup \rho\) and \(\rho^* = \rho \cup \sigma\). Let also \(\sigma^- = \{i \in \sigma \mid i-1 \notin \sigma\}\), \(\rho^- = \{i \in \rho \mid i-1 \notin \rho\}\).

**Lemma 3.** The sets \(\sigma^*, \rho^*, \sigma^-\) and \(\rho^-\) are finite or co-finite. Also, \(d(\sigma^*, \rho^*) \leq d(\sigma, \rho)\) and \(d(\sigma^-, \rho^-) \leq d(\sigma, \rho) + 1\).

**Lemma 4.** Let \(D\) be a Red \((\sigma, \rho)\)-dominating set of \(G\) and let \(u \in D\). The vertex \(u\) is a certificate for itself if and only if \(u \in \text{Blue}\) and \(|N_G(u) \setminus D| \in \sigma^+\). A vertex \(v \in N_G(u) \setminus \text{Blue}\) is a certificate for \(u\) if and only if \(D (\sigma^-, \rho^-)\)-dominates \(v\).

Let \(d \in \mathbb{N}\) and let \(A \subseteq V(G)\). Two red subsets \(X\) and \(Y\) of \(A\) are \(d\)-neighbor equivalent w.r.t. \(A\), denoted by \(X \equiv_A^d Y\), if \(\min(d, |X \cap N_G(x)|) = \min(d, |Y \cap N\) for all \(x \in A \cap \text{Blue}\). It is clear that \(\equiv_A^d\) is an equivalence relation and we let \(\text{neq} (\equiv_A^d)\) be its number of equivalence classes.
Lemma 5 ([1]). Let $d \in \mathbb{N}$ and let $A \subseteq V(G)$. Then $\text{neq}(\equiv^d_A) \preceq n^{d^2}$.

We will follow the same idea as in [4] where a minimum (or a maximum) $(\sigma, \rho)$-dominating set is computed. For every $i \in \{1, \ldots, n\}$ and every subset $X$ of $A_i \cap \text{Red}$, we denote by $\text{rep}_{p_i}^\sigma(X)$ the lexicographically smallest set $R \subseteq A_i \cap \text{Red}$ such that $|R|$ is minimised and $R \equiv^d_{p_i} X$. Notice that it can happen that $R = \emptyset$.

Lemma 6 ([4]). For every $i \in \{1, \ldots, n\}$, one can compute a list $LR_i$ containing all representatives w.r.t. $\equiv^d_{p_i}$ in time $O(\text{neq}(\equiv^d_{p_i}) \cdot \log(\text{neq}(\equiv^d_{p_i})) \cdot n^2)$. One can also compute a data structure that given a set $X \subseteq A_i \cap \text{Red}$ in time $O(\log(\text{neq}(\equiv^d_{p_i}))) \cdot |X| \cdot n$ allows us to find a pointer to $\text{rep}_{p_i}^\sigma(X)$ in $LR_i$. Similar statements hold for the list $LR_i$ containing all representatives w.r.t. $\equiv^d_{p_i}$.

Our goal now is to define a DAG, denoted by $\text{DAG}(G)$, the maximal paths of which correspond exactly to the 1-minimal Red $(\sigma, \rho)$-dominating sets.

For $1 \leq j \leq n$ and $C \subseteq A_j \cap \text{Blue}$ (or $C \subseteq A_j \cap \text{Red}$) we denote by $SG_j(C)$ (or by $\overline{SG}_j(C)$) the set $X$ obtained from $C$ if we initially set $X = C$ and recursively apply the following rule: let $x$ be the greatest (or smallest) vertex in $X$ such that $N(X \setminus \{x\}) \cap (\overline{A}_j \cap \text{Red}) = N(X) \cap (\overline{A}_j \cap \text{Red})$ (or $N(X \setminus \{x\}) \cap (A_j \cap \text{Red}) = N(X) \cap (A_j \cap \text{Red})$) and set $X = X \setminus \{x\}$. Notice that $SG_j(C)$ and $\overline{SG}_j(C)$ are both uniquely determined, and both have sizes bounded by $e$ from [1, Lemma 1]. Observe also that if $C \subseteq A_j \cap \text{Blue}$ (or $C \subseteq A_j \cap \text{Red}$), then $SG_j(C \cup \{x_i\}) = SG_j(SG_j(C) \cup \{x_i\})$ for all $\ell > j$ (or $\overline{SG}_j(C \cup \{x_i\}) = \overline{SG}_j(\overline{SG}_j(C) \cup \{x_i\})$ for all $\ell \leq j$).

Let $1 \leq j < n$ and let $(R_j, R'_j, C_j, C'_j) \in LR_j \times LR_j \times 2^{\overline{A}_j \cap \text{Blue}} \times 2^{A_j \cap \text{Blue}}$ and $(R_{j+1}, R'_{j+1}, C_{j+1}, C'_{j+1}) \in LR_{j+1} \times LR_{j+1} \times 2^{\overline{A}_{j+1} \cap \text{Blue}} \times 2^{A_{j+1} \cap \text{Blue}}$. There is an $\varepsilon$-arc-1 from $(R_j, R'_j, C_j, C'_j)$ to $(R_{j+1}, R'_{j+1}, C_{j+1}, C'_{j+1})$ if

1. $R_j \equiv^d_{p_j} R_{j+1}$ and $R'_j \equiv^d_{p_j} R'_{j+1}$, and
2. if $(x_{j+1} \notin \text{Blue})$ or $(x_{j+1} \in \text{Blue}$ and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \rho$ and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \rho^-$) then $(C_{j+1} = SG_j(C'_j)$ and $C'_{j+1} = \overline{SG}_j(C'_j)$, otherwise we should have $(|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \rho)$ and

1.1.a) if $N(x_{j+1}) \cap (A_j \cap \text{Red}) \neq \emptyset$, then $(C_{j+1} = SG_j(C_j \cup \{x_{j+1}\})$, else $C_{j+1} = \overline{SG}_j(C_j)$, and
1.1.b) if $N(x_{j+1}) \cap (A_j \cap \text{Red}) \neq \emptyset$, then $C'_{j+1} = SG_j(C_{j+1} \cup \{x_{j+1}\})$, else $C'_{j+1} = \overline{SG}_j(C_{j+1})$.

There is an $\varepsilon$-arc-2 from $(R_j, R'_j, C_j, C'_j)$ to $(R_{j+1}, R'_{j+1}, C_{j+1}, C'_{j+1})$ if

1. $R_{j+1} \equiv^d_{p_{j+1}} (R_j \cup \{x_{j+1}\})$, $R'_j \equiv^d_{p_j} (R'_{j+1} \cup \{x_{j+1}\})$, $x_{j+1} \in \text{Red}$, $(|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \in \sigma$ if $x_{j+1} \in \text{Blue}$, and
2. if $(x_{j+1} \notin \text{Blue})$ or $(x_{j+1} \in \text{Blue}$ and $|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \sigma)$, then $(C_{j+1} = SG_j(C'_j)$ and $C'_{j+1} = \overline{SG}_j(C'_{j+1})$, otherwise we should have $(|N(x_{j+1}) \cap (R_j \cup R'_{j+1})| \notin \sigma)$ and
(2.2.a) If \( N(x_{j+1} \cap (A_j + 1 \cap \text{Red}) \neq \emptyset \), then \( C_{j+1} = SG_j(C_j \cup \{x_{j+1}\}) \), else \( C_{j+1} = SG_j(C_j) \), and

(2.2.b) If \( N(x_{j+1} \cap (A_j \cap \text{Red}) \neq \emptyset \), then \( C'_j = SG_j(C'_j \cup \{x_{j+1}\}) \), else \( C'_j = SG_j(C'_j) \), and

(2.3) Either \( N(x_{j+1} \cap (C_j \cup C'_j) \neq \emptyset \) or \((x_{j+1} \in \text{Blue} \text{ and } |N(x_{j+1}) \cap (R_j \cup R'_j)| \in \sigma^*)\).

The nodes of \( \text{DAG}(G) \), \((R, R', C, C', i) \in LR_i \times LR_i \times 2^A, \text{Blue} \times 2^A, \text{Blue} \times [n])\) is a node of \( \text{DAG}(G) \) whenever \( x_i \in \text{Red} \), \( C = SG_j(C) \) and \( C' = SG_j(C') \). We call \( i \) the index of \((R, R', C, C', i)\). Finally \( s = (\emptyset, \emptyset, \emptyset, \emptyset, 0) \) is the source node and \( t = (\emptyset, \emptyset, \emptyset, \emptyset, n + 1) \) is the terminal node of \( \text{DAG}(G) \).

The arcs of \( \text{DAG}(G) \). There is an arc from the node \((R_0, R_0', C_0, C_0', j) \in LR_0 \times LR_0 \times 2^A, \text{Blue} \times 2^A, \text{Blue} \times [n])\) to the node \((R_1, R_1', C_1, C_1', j + p) \in LR_1 \times LR_1 \times 2^A, \text{Blue} \times 2^A, \text{Blue} \times [n])\) with \( 1 \leq j < j + p \leq n \) if there exist tuples \((R_i, R'_i, C_i, C'_i), \ldots, (R_{p-1}, R'_{p-1}, C_{p-1}, C'_{p-1})\) such that (1) for each \( 1 \leq i < p \), \((R_i, R'_i, C_i, C'_i) \in LR_i \times LR_i \times 2^A, \text{Blue} \times 2^A, \text{Blue} \times [n])\) and there is an \( \varepsilon \)-arc-1 from \((R_i, R'_i, C_i, C'_i)\) to \((R_i, R'_i, C_i, C'_i)\), and (2) there is an \( \varepsilon \)-arc-2 from \((R_{p-1}, R'_{p-1}, C_{p-1}, C'_{p-1})\) to \((R_p, R'_p, C_p, C'_p)\).

There is an arc from the source node to a node \((R, R', C, C', j) \in \text{DAG}(G) \) if \((S = \{x \in (A_j \cap \text{Blue}) \cup \{x_j\}, N(x) \cap (A_j \cap \text{Red}) \neq \emptyset \text{ and } |N(x) \cap (\{x_j\} \cup R')| \in \rho^*\})\)

(S1) \( \{x_j\} = A_j \text{ and } (\{x_j\} \cup R')\) \( (\sigma, \rho) \)-dominates \( A_j \cap \text{Blue} \),

(S2) If \( x_j \in \text{Blue} \) and \( |N(x_j) \cap R'| \in \sigma^* \) then \( C = SG_j(S \cup \{x_j\}) \), otherwise \( C = SG_j(S) \), and

(S3) Either \((N(x_j) \cap (C' \cup C) \neq \emptyset \) or \((x_j \in \text{Blue} \text{ and } |N(x_j) \cap R'| \in \sigma^*\).

There is an arc from a node \((R, R', C, C', j) \in \text{DAG}(G) \) to the terminal node if

(T1) \(|N(x) \cap R'| \in \rho \text{ for each } x \in A_j + 1 \cap \text{Blue} \text{, and} \)

(T2) \( C' = SG_j(\{x \in A_j \cap \text{Blue} \text{ | } N(x) \cap (A_j \cap \text{Red}) \neq \emptyset \text{ and } |N(x) \cap R'| \in \rho^*\}). \)

If \( P = (s, v_1, v_2, \ldots, v_p, t) \) is a path in \( \text{DAG}(G) \), then the trace of \( P \), denoted by \( \text{trace}(P) \), is defined as \( \{x_{j_1}, x_{j_2}, \ldots, x_{j_p}\} \) where for all \( i \in \{1, 2, \ldots, p\} \), \( j_i \) is the index of the node \( v_i \). We have the following lemmas.

**Lemma 7.** Let \( P \) be the set of paths in \( \text{DAG}(G) \) from the source node to the terminal node. The mapping which associates with every \( P \in P \) \( \text{trace}(P) \) is a one-to-one correspondence with the set of \( 1 \)-minimal \( \text{Red} \) \( (\sigma, \rho) \)-dominating sets.

**Lemma 8.** \( \text{DAG}(G) \) is a DAG and can be constructed in time \( O(n^{c-d}) \).

We can now prove Theorem 1. By Lemma 8 \( \text{DAG}(G) \) is a DAG and can be constructed in time \( O(n^{c-d}) \). By Lemma 7 it is sufficient to count and enumerate the maximal paths in \( \text{DAG}(G) \), and since we can count the maximal paths and enumerate them with linear delay (see for instance [15]), this concludes the proof.
4 Enumeration of minimal dominating sets for unit square graphs

Let $G$ be a unit square graph and suppose that $f : V(G) \to \mathbb{Q}^2$ is a realization of $G$. (See Section 2 for more details on the point model of unit square graphs used in our paper.) For a vertex $v \in V(G)$, $\frac{f(v)}{||f(v)||}$ is the fractional part of the $x$-coordinate of the point representing $v$. Let $v_1, \ldots, v_n$ be a linear ordering of the vertex set of $G$ such that $\frac{f(v_i)}{||f(v_i)||} \leq \frac{f(v_j)}{||f(v_j)||}$ for all $j > i$. We prove that the MIM-width of $v_1, \ldots, v_n$ is bounded by $O(\text{diam}^2)$ where $\text{diam}$ is the diameter of $G$, which we state in the following.

**Theorem 2.** For a unit square graph $G$, $u \in V(G)$ and a positive integer $r$, $\text{liminw}(G[N_G[u]]) = O(r^2)$. Moreover, if a realization $f : V(G) \to \mathbb{Q}^2$ of $G$ is given, then a linear ordering of vertices of MIM-width $O(r^2)$ can be constructed in polynomial time.

We will now explain how to use this property and Theorem 1, to obtain an incremental polynomial time enumeration algorithm for the minimal dominating sets of $G$. To do it, we use a variant of the flipping method proposed in [10].

Given a minimal dominating set $D^*$, the flipping operation replaces an isolated vertex of $G[D^*]$ with its neighbor outside of $D^*$, and, if necessary, adds or deletes some vertices to obtain new minimal dominating sets $D$, such that $G[D]$ has more edges compared to $G[D^*]$. The enumeration algorithm starts with enumerating all maximal independent sets of the input graph $G$ using the algorithm of Johnson, Papadimitriou, and Yannakakis [12], which gives the initial minimal dominating sets. Then the flipping operation is applied to every appropriate minimal dominating set found, to find new minimal dominating sets inducing subgraphs with more edges.

Let $G$ be a graph. Let also $D \subseteq V(G)$. For $u \in D$, $C_D[u] = \{v \in V(G) \mid v \in N_G[u] \setminus N_G[D \setminus \{u\}]\}$ and $C_D(u) = \{v \in V(G) \mid v \in N_G(u) \setminus N_G[D \setminus \{v\}]\} = C_D[u] \setminus \{u\}$. Observe that if $D$ is a minimal dominating set, then $C_D[u]$ is the set of certificates for a vertex $u \in D$.

Let us describe the variant of the flipping operation from [10], that we use. Let $G$ be the input graph; we fix an (arbitrary) order of its vertices: $v_1, \ldots, v_n$. Suppose that $D'$ is a dominating set of $G$. We say that the minimal dominating set $D$ is obtained from $D'$ by greedy removal of vertices (with respect to order $v_1, \ldots, v_n$) if we initially let $D = D'$, and then recursively apply the following rule: If $D$ is not minimal, then find a vertex $v_i$ with the smallest index $i$ such that $D \setminus \{v_i\}$ is a dominating set in $G$, and set $D = D \setminus \{v_i\}$. Clearly, when we apply this rule, we never remove vertices of $D'$ that have certificates. Whenever greedy removal of vertices of a dominating set is performed, it is done with respect to this ordering.

Let $D$ be a minimal dominating set of $G$ such that $G[D]$ has at least one edge $uw$. Then the vertex $u \in D$ is dominated by the vertex $w \in D$. Therefore, $C_D[u] = C_D(u) \neq \emptyset$. Let $X$ be a non-empty inclusion-maximal independent set such that $X \subseteq C_D(u)$. Consider the set $D' = (D \setminus \{u\}) \cup X$. Notice that
$D'$ is a dominating set in $G$, since all vertices of $C_D(u)$ are dominated by $X$ by the maximality of $X$ and $u$ is dominated by $w$, but $D'$ is not necessarily minimal, because it can happen that $X$ dominates all the certificates of some vertex of $D\setminus\{u\}$. We apply greedy removal of vertices to $D'$ to obtain a minimal dominating set. Let $Z$ be the set of vertices that are removed by this to ensure minimality. Observe that $X \cap Z = \emptyset$ and $u \notin Z$ by the definition of these sets; in fact there is no edge between a vertex of $X$ and a vertex of $Z$. Finally, let $D^* = ((D\setminus\{u\}) \cup X) \setminus Z$.

It is important to notice that $|E(G[D^*])| < |E(G[D])|$. Indeed, to construct $D^*$, we remove the endpoint $u$ of the edge $uw \in E(G[D])$ and, therefore, reduce the number of edges. Then we add $X$ but these vertices form an independent set in $G$ and, because they are certificates for $u$ with respect to $D$, they are not adjacent to any vertex of $D\setminus\{u\}$. Therefore, $|E(G[D^*])| \leq |E(G[D^1])| < |E(G[D])|$.

The flipping operation is exactly the reverse of how we generated $D^*$ from $D$; i.e., it replaces a non-empty independent set $X \subseteq G[D^*] \cap N_G(u)$ for a vertex $u \notin D^*$ with their neighbor $u$ in $G$ to obtain $D$. In particular, we are interested in all minimal dominating sets $D$ that can be generated from $D^*$ in this way. Given $D$ and $D^*$ as defined above, we say that $D^*$ is a parent of $D$ with respect to flipping $u$ and $X$. We say that $D^*$ is a parent of $D$ if there is a vertex $u \in V(G)$ and an independent set $X \subseteq N_G(u)$ such that $D^*$ is a parent with respect to flipping $u$ and $X$. It is important to note that each minimal dominating set $D$ such that $E(G[D]) \neq \emptyset$ has a unique parent with respect to flipping of any $u \in D \cap N_G[D\setminus\{u\}]$ and a maximal independent set $X \subseteq C_D(u)$, as $Z$ is lexicographically selected by a greedy algorithm. Similarly, we say that $D$ is a child of $D^*$ (with respect to flipping $u$ and $X$) if $D^*$ is the parent of $D$ (with respect to flipping $u$ and $X$). The proof of the following lemma is implicit in [10].

**Lemma 9 ([10]).** Suppose that for a graph $G$, all independent sets $X \subseteq N_G(u)$ for a vertex $u$ can be enumerated in polynomial time. Suppose also that there is an enumeration algorithm $A$ that, given a minimal dominating set $D^*$ of a graph $G$ such that $G[D^*]$ has an isolated vertex, a vertex $u \in V(G) \setminus D^*$ and a non-empty independent set $X$ of $G[D^*]$ such that $X \subseteq D^* \cap N_G(u)$, generates with polynomial delay a family of minimal dominating sets $D$ with the property that $D$ contains all minimal dominating sets $D$ that are children of $D^*$ with respect to flipping $u$ and $X$. Then all minimal dominating sets of $G$ can be enumerated in incremental polynomial time.

To obtain our main result, we will show that there is indeed an algorithm as algorithm $A$ described in the statement of Lemma 9 when the input graph $G$ is a unit square graph. We show that we can construct $A$ by reduction to the enumeration of minimal Red dominating sets in an auxiliary colored induced subgraph of $G[N_3^G[u]]$. Let $D^*$ be a minimal dominating set of a graph $G$ such that $G[D^*]$ has an isolated vertex. Let also $u \in V(G) \setminus D^*$ and $X$ is a non-empty independent set of $G[D^*]$ such that $X \subseteq D^* \cap N_G(u)$. Consider the
set $D' = (D \setminus X) \cup \{u\}$. Denote by Blue the set of vertices that are not dominated by $D'$. Notice that Blue $\subseteq N_G(X) \setminus N_G[u]$. Therefore, Blue $\subseteq N_G^2[u]$. Let Red $= N_G(Blue) \setminus N_G[X]$. Clearly, Red $\subseteq N_G^2[u]$. We construct the colored graph $H = G[Red \cup Blue]$. Let $\mathcal{A}'$ be an algorithm that enumerates minimal Red dominating sets in $H$. Assume that if Blue $= \emptyset$, then $\mathcal{A}'$ returns $\emptyset$ as the unique Red dominating set. We construct $\mathcal{A}$ as follows.

Step 1. If $\mathcal{A}'$ returns an empty list of sets, then $\mathcal{A}$ returns an empty list as well.

Step 2. For each Red dominating set $R$ of $H$, consider $D_2 = D_1 \setminus R$ and construct a minimal dominating set $D$ from $D_2$ by greedy removal.

**Lemma 10.** If $\mathcal{A}'$ lists all minimal Red dominating sets with polynomial delay, then $\mathcal{A}$ generates with polynomial delay a family of minimal dominating sets $\mathcal{D}$ with the property that $\mathcal{D}$ contains all minimal dominating sets $\mathcal{D}$ that are children of $\mathcal{D}$ with respect to flipping $u$ and $X$.

Now we are ready to prove the main result of the section.

**Theorem 3.** For a unit square graph $G$ given with its realization $f$, all minimal dominating sets of $G$ can be enumerated in incremental polynomial time.

**Proof.** It is straightforward to observe that for a vertex $u$ of a unit square graph $G$, any independent set $X \subseteq N_G(u)$ has at most 4 vertices. Hence, all independent sets $X \subseteq N_G(u)$ for a vertex $u$ can be enumerated in polynomial time. By combining Theorems 1 and 2, and Lemmas 9 and 10, we obtain the claim. □

**References**

1. R. Belmonte, M. Vatshelle. Graph classes with structured neighborhoods and algorithmic applications. *Theor. Comput. Sci.* 511, 54–65 (2013).
2. A. Brandstädt, V.B. Le, J.P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA (1999).
3. H. Breu. Algorithmic aspects of constrained unit disk graphs. PhD thesis, The University of British Columbia (1996).
4. B.M. Bui-Xuan, J.A. Telle, M. Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theor. Comput. Sci.* 511, 66–76 (2013).
5. L.S. Chandran, M.C. Francis, N. Sivadasan. On the cubicity of interval graphs. *Graphs and Combinatorics* 25(2), 169–179 (2009).
6. B. Courcelle. Linear delay enumeration and monadic second-order logic. *Discrete Applied Mathematics* 157, 2675–2700 (2009).
7. T. Eiter, G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Comput.* 24, 1278–1304 (1995).
8. T. Eiter, G. Gottlob. Hypergraph transversal computation and related problems in Logic and AI. Proceedings of *JELIA 2002*, LNCS 2424, pp. 549–564 (2002).
9. T. Eiter, G. Gottlob, K. Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Comput.* 32, 514–537 (2003).
10. P. A. Golovach, P. Heggernes, D. Kratsch, Y. Villanger. An incremental polynomial time algorithm to enumerate all minimal edge dominating sets. *Algorithmica* 72, 836-859 (2015).
11. P. A. Golovach, P. Heggernes, M. Kante, D. Kratsch, Y. Villanger. Enumerating minimal dominating sets in chordal bipartite graphs. *Discrete Applied Mathematics*, to appear. DOI 10.1016/j.dam.2014.12.010.
12. D.S. Johnson, C.H. Papadimitriou, M. Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.* 27(3), 119–123 (1988).
13. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine. On the enumeration of minimal dominating sets and related notions. *SIAM J. Discrete Math.*, 28, 1916–1929 (2014).
14. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine. On the Neighbourhood Helly of some Graph Classes and Applications to the Enumeration of Minimal Dominating Sets. Proceedings of *ISAAC 2012*, LNCS 7676, pp. 289–298 (2012).
15. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, T. Uno., On the Enumeration and Counting of Minimal Dominating sets in Interval and Permutation Graphs. Proceedings of *ISAAC 2013*, LNCS 8283, pp. 339-349 (2013).
16. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, T. Uno. A Polynomial Delay Algorithm for Enumerating Minimal Dominating Sets in Chordal Graphs. Proceedings of *WG 2015*, to appear. arxiv:1407.2036 (2014).
17. M. M. Kanté, V. Limouzy, A. Mary, L. Nourine, T. Uno. Polynomial Delay Algorithm for Listing Minimal Edge Dominating sets in Graphs. Proceedings of *WADS 2015*, LNCS 9214, pp. 446-357 (2015).
18. L. Khachiyan, E. Boros, K. Borys, K. M. Elbassioni, V. Gurvich. Generating all vertices of a polyhedron is hard. *Discrete & Computational Geometry* 39, 174–190 (2008).
19. L. Khachiyan, E. Boros, K. M. Elbassioni, V. Gurvich. On enumerating minimal dicuts and strongly connected subgraphs. *Algorithmica* 50, 159–172 (2008).
20. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM J. Comput.* 9, 558–565 (1980).
21. I. Rauf. Polynomially Solvable Cases of Hypergraph Transversal and Related Problems. PhD thesis, Saarland University (2011).
22. R. E. Tarjan. Enumeration of the elementary circuits of a directed graph. *SIAM J. Comput.* 2, 211–216 (1973).
23. J.A. Telle, A. Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.* 10(4), 529–550 (1997).
24. M. Vatshelle. New width parameters of graphs. PhD thesis, University of Bergen (2012).