Enumerating Minimal Dominating Sets in Chordal Bipartite Graphs*

Petr A. Golovach† Pinar Heggernes† Mamadou M. Kanté‡
Dieter Kratsch§ Yngve Villanger†

Abstract

We show that all minimal dominating sets of a chordal bipartite graph can be generated in incremental polynomial, hence output polynomial, time. Enumeration of minimal dominating sets in graphs is equivalent to enumeration of minimal transversals in hypergraphs. Whether the minimal transversals of a hypergraph can be enumerated in output polynomial time is a well-studied and challenging question that has been open for several decades. With this result we contribute to the known cases having an affirmative reply to this important question.

1 Introduction

Enumeration in graphs is a central area of graph algorithms, and it involves generating or listing all vertex or edge subsets of a graph satisfying a given property. As the number of objects to be enumerated is usually exponential in the size of the input graph, tractability of an enumeration problem is defined based on the size of the output. In particular, if an enumeration algorithm runs in time that is polynomial in the size of the input graph plus the number of enumerated objects, then it is called output polynomial. A large amount of results have been dedicated to output polynomial enumeration algorithms, e.g., [1, 2, 10, 11, 12, 18, 24, 25, 26, 28, 32, 33, 34], and for various enumeration problems it has been shown that no output polynomial time algorithm can exist unless P = NP [24, 26, 28].

A famous and well-studied enumeration problem is that of enumerating the minimal hitting sets of the hyperedges, called minimal transversals, of a hypergraph

*The research leading to these results has received funding from the Research Council of Norway, the French National Research Agency (DORSO and AGAPE projects), and the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 267959.

†University of Bergen, Norway, {petr.golovach, pinar.heggernes, yngve.villanger}@ii.uib.no
‡Clermont-Université, Université Blaise Pascal, LIMOS, CNRS, Aubiére, France, mamadou.kante@isima.fr
§Université de Lorraine, Metz, France, kratsch@univ-metz.fr.

[12, 13]. Whether the minimal transversals of a hypergraph can be enumerated in output polynomial time has been identified as a fundamental open question by several authors [10, 11, 12, 13, 14, 18, 31], and it has remained unresolved for the last 30 years. It has recently been shown that enumerating the minimal transversals of a hypergraph is equivalent to enumerating the minimal (total) dominating sets of a graph [19]. This has enabled attacking the mentioned enumeration problem in new ways, as (total) dominating sets form one of the best studied notions in computer science [17].

Since the tractability of enumerating the minimal transversals of a hypergraph, or equivalently, the minimal dominating sets of a graph, has been open for so long, several tractable special cases have been identified [4, 5, 8, 9, 10, 12, 13, 29, 30]. In particular, output polynomial algorithms to enumerate minimal dominating sets in graphs exist for graphs of bounded treewidth and of bounded clique-width [7], interval graphs [10], strongly chordal graphs [10], chordal graphs [22], planar graphs [12], degenerate graphs [12], split graphs [19], path graphs [20], permutation graphs [21], line graphs [15, 20, 23], and graphs of girth at least 7 [15].

Table 1 summarizes for well-known classes of graphs the best known output polynomial time algorithm to enumerate all minimal dominating sets. The definitions of incremental polynomial (IncP), polynomial delay (DelayP) and linear delay (DelayL) are given in Section 2. (Total P) indicates output polynomial.

Graph Classes	TotalP	IncP	DelayP	DelayL
degenerate graphs			[12]	
$\log(n)$-degenerate graphs	[12]			
bounded conformality		[3]		
line graphs		[15]	[23]	
chordal graphs			[22]	
$(C_{\leq 6})$-free graphs		[15]		
$(C_4, C_5, claw)$-free graphs		[20]		
split graphs				[19]
P_6-free chordal graphs				[19]
strongly chordal graphs			[10]	
permutation graphs				[20]
interval graphs				[20]
bounded clique-width graphs				[7]
chordal bipartite graphs		[*]		

Table 1: Algorithms to enumerate all minimal dominating sets. ([*] this paper)

Based on the supergraph technique for enumerating vertex subsets in graphs [2, 23, 25, 32, 34], Golovach et al. [15] presented a flipping method to generate the out-neighbors of a node of the supergraph, i.e., to generate new minimal dominating sets from a parent dominating set. Using this flipping method they were able to give output polynomial algorithms for enumerating the minimal dominating sets of line graphs and graphs of large girth.

In this paper we show that the mentioned flipping method can be used on chordal bipartite graphs as well. In particular, we show that the minimal dominating sets of a chordal bipartite graph can be enumerated in incremental polynomial, hence output polynomial, time. To obtain our result we first show how to enumerate with polynomial delay the minimal total dominating sets of a chordal bipartite graph. This first part does not involve the flipping method and can be of independent interest. In particular, a direct consequence of our algorithm is another polynomial delay algorithm for enumerating the minimal transversals of totally balanced hypergraphs [6]. Chordal bipartite graphs are graphs that are weakly chordal and bipartite. They form a well studied graph class [6], as they mimic in bipartite graphs the behavior of the more famous class of chordal graphs. It should be noted that the complexity of enumerating all minimal dominating sets in weakly chordal graphs is the same as the complexity of enumerating all minimal transversals in hypergraphs [19]. On the other hand, it is an open question whether all minimal dominating sets of a bipartite graph can be enumerated in output polynomial time; same holds for unit disk graphs.

2 Definitions and notation

We consider finite undirected graphs without loops or multiple edges. Let $G = (V, E)$ be such a graph. The subgraph of G induced by a subset $U \subseteq V$ is denoted by $G[U]$. We write $G - U$ for $G[V \setminus U]$; if $U = \{u\}$, we simply write $G - u$. For a vertex v, we denote by $N_G(v)$ its *(open) neighborhood*, that is, the set of vertices that are adjacent to v. The *closed neighborhood* of v is the set $N_G(v) \cup \{v\}$, and it is denoted by $N_G[v]$. If $N_G(v) = \emptyset$ then v is *isolated*, and if v has exactly one neighbor, it is *pendant*. For a set $U \subseteq V$, $N_G[U] = \cup_{v \in U} N_G[v]$, and $N_G(U) = N_G[U] \setminus U$. We may omit subscripts when there is no confusion. G is a *bipartite* graph if its set of vertices can be partitioned into two sets V_1 and V_2 in such a way that for any edge $e \in E$, one endpoint of e is in V_1 and the other is in V_2. In this case we write $G = (V_1, V_2, E)$. An edge $uv \in E$ is *bisimplicial* if the subgraph induced by $N(u) \cup N(v)$ is a complete bipartite graph.

A vertex v *dominates* a vertex u if $u \in N[v]$; similarly v dominates a set of vertices U if $U \subseteq N[v]$. For two sets $D, U \subseteq V$, D dominates U if $U \subseteq N[D]$. A set of vertices D is a *dominating set* of $G = (V, E)$ if D dominates V, and D is a *total dominating set* if each vertex of G has a neighbor in D. A (total) dominating set is *minimal* if no proper subset of it is a (total) dominating set. Let D be a dominating set of G, and let $v \in D$. Vertex u is a *private vertex*, or simply *private*, for a vertex v (with respect to D) if u is dominated by v but is not dominated by $D \setminus \{v\}$. Clearly, a dominating set D is a minimal dominating set if and only if each vertex of D has a private vertex. We denote by $P_D[v]$ the set of all private vertices for v; a vertex of D can be private for itself. Vertex u is a *private neighbor* of $v \in D$ if $u \in N(v) \cap P_D[v]$. The set of all private neighbors of v is denoted by $P_D(v)$. Note that $P_D[v] = P_D(v) \cup \{v\}$ if v is isolated in $G[D]$, and otherwise

$P_D[v] = P_D(v)$.

For the definition of *red-blue domination*, consider a bipartite graph $G = (R, B, E)$. We refer to the vertices of R as the *red* vertices, the vertices of B as the *blue* vertices, and we say that G is a *red-blue graph.* A set of vertices $D \subseteq R$ is a *red dominating set* if D dominates B, and D is *minimal* if no proper subset of it dominates B. As above, we say that a vertex $u \in B$ is a *private vertex*, or simply *private*, for a vertex $v \in D$ if u is dominated by v but is not dominated by $D \setminus \{v\}$. It is straightforward to see that $D \subseteq R$ is a minimal red dominating set if and only if D dominates B and each vertex of D has a private vertex. Similarly, we define *(minimal) blue dominating sets* as (minimal) subsets of B that dominate R.

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 $delay(\mathcal{A}, 1) = T(\mathcal{A}, 1)$ and $delay(\mathcal{A}, i) = T(\mathcal{A}, i) - T(\mathcal{A}, i - 1)$. The *delay* of $\mathcal{A}$ is $\max\{delay(\mathcal{A}, i)\}$. Algorithm $\mathcal{A}$ runs in *incremental polynomial* time if there is a polynomial $p(x, i)$ such that $delay(\mathcal{A}, i) \leq p(n + m, i)$. 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 $delay(\mathcal{A}, 1)$ is bounded by a polynomial in $n + m$ and $delay(\mathcal{A}, i)$ is bounded by a linear function in $n + m$.

A graph is a *chordal bipartite graph* if it is bipartite and every cycle of length at least 6 has a chord, i.e., an edge between two non-consecutive vertices of the cycle. Chordal bipartite graphs were introduced by Golumbic and Goss [16] as a natural bipartite analogue of chordal graphs. We need the following property of chordal bipartite graphs.

Theorem 1 ([16])**.** *Every non-trivial chordal bipartite graph contains a bisimplicial edge. Moreover, a bipartite graph is chordal bipartite if and only if every non-trivial induced subgraph of it has a bisimplicial edge.*

3 Enumeration of minimal red dominating sets in red-blue chordal bipartite graphs

In this section we give a polynomial delay algorithm for enumerating the minimal red dominating sets of a red-blue chordal bipartite graph. This result is needed for the next section, and of independent interest, it implies that the minimal total dominating sets of a chordal bipartite graph can be enumerated with polynomial delay.

Theorem 2. *All minimal red dominating sets of a red-blue chordal bipartite graph with n vertices and m edges can be enumerated with delay $O(n \cdot \min\{m \log n, n^2\})$.*

Proof. Let $G = (R, B, E)$ be a chordal bipartite graph. We describe a recursive branching algorithm that achieves the statement of the theorem.

If $B = \emptyset$, then $\emptyset$ is the unique minimal red dominating set. Observe that G has no minimal red dominating set if and only if B has an isolated vertex. Consequently, if G has an isolated blue vertex, we output an empty list. For the rest of the proof we assume that $B \neq \emptyset$ and there is no isolated vertex in B.

The algorithm takes as input a set of red vertices D and a red-blue induced subgraph $H = (R', B', E')$ of G such that $B' \neq \emptyset$ and there is no isolated vertex in B', $D \cap R' = \emptyset$, and the vertices of B' are not adjacent to the vertices of D in G. Initially we call the algorithm with $D = \emptyset$ and $H = G$.

Since $B' \neq \emptyset$ and has no isolated blue vertex, H has an edge, and we find a bisimplicial edge uv with $u \in R'$ and $v \in B'$, which exists by Theorem 1. Then we branch in two directions for u, where in the one branch we generate all the minimal red dominating sets containing u, and in the other branch all of those that do not contain u. This is done in two recursive calls: in the first we include u into the current red dominating set suggestion D, and in the second we exclude it from such a possibility for the rest of the execution of the algorithm. In each of the cases, we check to make sure that we can obtain a minimal red dominating set by this selection before we proceed with the recursive calls. If we cannot, or if we already have found a minimal red dominating set, the recursion stops.

Branch 1: include u. If $N_H(u) = B'$, then output $D \cup \{u\}$. Otherwise, if $H' = H - (N_H(u) \cup N_H(v))$ has no isolated blue vertex, then call the algorithm with input $(D \cup \{u\},\ H')$.

Branch 2: exclude u. If $H - u$ has no isolated blue vertex, then call the algorithm with input $(D,\ H - u)$.

Let us prove the correctness of this algorithm. Assume that $B' \neq \emptyset$, H has no isolated blue vertex, all the vertices of $B \setminus B'$ are dominated by D, and each vertex of D has a private that is not adjacent to the vertices of R'. These conditions are satisfied trivially in the beginning, as $H = G$ and $D = \emptyset$. By these conditions, H has a minimal red dominating set D', and $D' \subseteq R'$ is a minimal red dominating set of H if and only if $D \cup D'$ is a minimal red dominating set of G. Clearly, for each minimal red dominating set D' of H, either $u \in D'$ or $u \notin D'$.

If $N_H(u) = B'$, then $D' = \{u\}$ is the unique minimal dominating set of H that contains u. Suppose that H has a minimal red dominating set $D' \neq \{u\}$ such that $u \in D'$. Notice that $B'' = B' \setminus N_H(u) \neq \emptyset$ in this case. For any $w \in N_H(v)$, we have that $N_H(u) \subseteq N_H(w)$ since uv is a bisimplicial edge. Consequently, $(N_H(v) \setminus \{u\}) \cap D' = \emptyset$ as otherwise u would have no privates and, therefore, $D'' = D' \setminus \{u\} \subseteq R' \setminus N_H(v)$. Because D' dominates B', D'' dominates B''. Moreover, because D' is a minimal red dominating set of H, D'' is a minimal set that dominates B''. Hence, D'' is a minimal red dominating set of H'. Notice that because H' has a minimal red dominating set, H' has no isolated blue vertex. Assume now that $D'' \neq \emptyset$ is a minimal red dominating set of H'. Then $B'' = B' \setminus N_H(u) \neq \emptyset$ and H' has no isolated blue vertex. Also $D' = D'' \cup \{u\}$ is a red dominating set of H. We have that $N_H(v) \cap D'' = \emptyset$. Hence, v is not dominated in H by any vertex of D'' and, therefore,

v is a private for u with respect to D'. Any vertex $w \in D''$ has its private w' in B'' with respect to D''. Since $w' \in B''$, $w' \notin N_H(u)$, i.e., w' is a private for w with respect to D'. We obtain that D' is a minimal red dominating set of H. It follows that in Branch 1 we enumerate all minimal red dominating sets of H that include u.

If $H-u$ has an isolated blue vertex, then any minimal red dominating set of H contains u. Also D' is a minimal red dominating set of H that does not include u if and only if D' is a minimal red dominating set of $H-u$. It follows that in Branch 2 we enumerate all minimal red dominating sets of H that exclude u.

To complete the correctness proof, observe that our algorithm constructs a search tree every leaf of which corresponds to a distinct minimal red dominating set.

It remains to evaluate the running time. A bisimplicial edge can be found in time $O(\min\{m \log n, n^2\})$ by the results of Kloks and Kratsch [27]. Then H' can be constructed and we can verify that H' has no isolated blue vertex in time $O(n+m)$. Because the depth of the search tree is at most n, then the delay is bounded by $O(n \cdot \min\{m \log n, n^2\})$. □

In the next section, we will use Theorem 2 in the construction of an algorithm for enumerating all minimal dominating sets of a chordal bipartite graph. As a separate result, Theorem 2 implies that all minimal total dominating sets of chordal bipartite graphs can be enumerated with polynomial delay.

Corollary 1. *All minimal total dominating sets of a chordal bipartite graph with n vertices and m edges can be enumerated with delay $O(n \cdot \min\{m \log n, n^2\})$.*

Proof. Assume that $G = (R, B, E)$ is a red-blue chordal bipartite graph. We claim that D is a minimal total dominating set of G if and only if $D_R = D \cap R$ is a minimal red dominating set and $D_B = D \cap B$ is a minimal blue dominating set. To see it, it is sufficient to observe that because R and B are independent sets, D is a total dominating set of G if and only if D_R dominates B and D_B dominates R, i.e., D_R is a red dominating set and D_B is a blue dominating set.

By this claim , we can enumerate all minimal total dominating sets as follows. We use Theorem 2 to enumerate all minimal red dominating sets, and then for each minimal red dominating set, we enumerate all minimal blue dominating sets and output all the unions of this minimal red dominating set and all blue dominating sets. □

4 Enumeration of minimal dominating sets of chordal bipartite graphs

In this section we show that the minimal dominating sets of chordal bipartite graphs can be enumerated in incremental polynomial time. We use the flipping method proposed by Golovach, Heggernes, Kratsch, and Villanger in [15]. Given a minimal dominating set D^*, the flipping operation replaces an isolated vertex of $G[D^*]$ with a 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

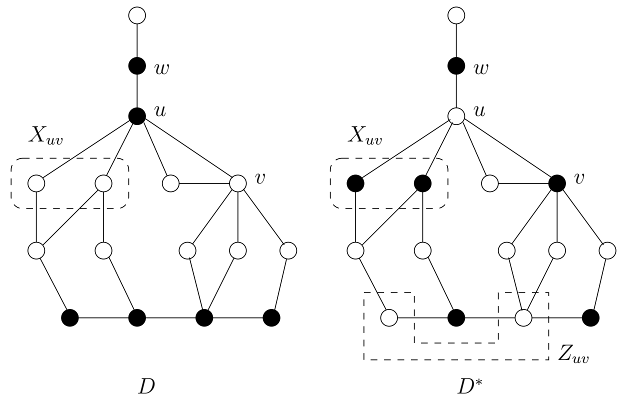

Figure 1: [15] A minimal dominating set D and its parent D^*; the vertices of D and D^* are black.

with enumerating all maximal independent sets of the input graph G using the algorithm of Johnson, Papadimitriou, and Yannakakis [18], which gives the initial set of 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 us describe the flipping operation of [15]. 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 private neighbors. 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 vertex $u \in D$ is dominated by vertex $w \in D$. Let $v \in P_D(u)$. Let $X_{uv} \subseteq P_D(u) \setminus N[v]$ be a maximal independent set in $G[P_D(u) \setminus N[v]]$ selected greedily with respect to ordering $v_1, \ldots, v_n$, i.e., we initially set $X_{uv} = \emptyset$ and then recursively include in X_{uv} the vertex of $P_D(u) \setminus (N[\{v\} \cup X_{uv}])$ with the smallest index as long as it is possible. Consider the set $D' = (D \setminus \{u\}) \cup X_{uv} \cup \{v\}$. Notice that D' is a dominating set in G, since all vertices of $P_D(u)$ are dominated by $X_{uv} \cup \{v\}$, but D' is not necessarily minimal, because it can happen that $X_{uv} \cup \{v\}$ dominates all the privates of some vertex of $D \setminus \{u\}$. We apply greedy removal of vertices to D' to obtain a minimal dominating set. Let Z_{uv} be the set of vertices that are removed by this to ensure minimality. Observe that $X_{uv} \cap Z_{uv} = \emptyset$ by the definition of these sets; in fact there is no edge between a vertex of X_{uv} and a vertex of Z_{uv}. Finally, let $D^* = ((D \setminus \{u\}) \cup X_{uv} \cup \{v\}) \setminus Z_{uv}$; see Figure 1 from [15] for an example.

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 $X_{uv} \cup \{v\}$ but these vertices compose an independent set in G and, because they are privates for u with respect to D, they are not adjacent to any vertex of $D \setminus \{u\}$. Therefore, $|E(G[D^*])| \leq |E(G[D'])| < |E(G[D])|$.

The *flipping* operation is exactly the *reverse* of how we generated D^* from D; i.e., it replaces an isolated vertex v of $G[D^*]$ with a 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 v*. We say that D^* is a *parent* of D if there are vertices $u, v \in V(G)$ such that D^* is a parent with respect to flipping u and v. 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 vertices $u \in D \cap N[D \setminus \{u\}]$ and $v \in P_D(u)$, as both sets X_{uv} and Z_{uv} are lexicographically first sets selected by a greedy algorithm. Similarly, we say that D is a *child* of D^* (with respect to flipping u and v) if D^* is the parent of D (with respect to flipping u and v).

The description of the flipping operation is complete. Golovach, Heggernes, Kratsch and Villanger [15] proved the following lemma.

Lemma 1 ([15])**.** *Suppose that there is an enumeration algorithm $\mathcal{A}$ that, given a minimal dominating set D^* of a graph G with n vertices and m edges such that $G[D^*]$ has isolated vertices, an isolated vertex v of $G[D^*]$, and a neighbor u of v in G, generates with delay $O(p(n, m))$ a family of minimal dominating sets $\mathcal{D}$ with the property that $\mathcal{D}$ contains all minimal dominating sets D that are children of D^* with respect to flipping u and v. Then all minimal dominating sets of G can be enumerated with delay $O((p(n, m) + n^2)m|\mathcal{L}|^2)$ and total running time $O((p(n, m)+n^2)m|\mathcal{L}^*|^2)$, where $\mathcal{L}$ is the family of already generated minimal dominating sets and $\mathcal{L}^*$ is the family of all minimal dominating sets.*

To obtain our main result, we will show that there is indeed an algorithm as algorithm $\mathcal{A}$ described in the statement of Lemma 1 when the input graph G is chordal bipartite. We first need the following observation about Z_{uv} with respect to flipping u and v to obtain a child D from a parent D^*; it states that no vertex in Z_{uv} is private for itself with respect to D.

Observation 1. *If $D = (D^* \setminus (X_{uv} \cup \{v\})) \cup (Z_{uv} \cup \{u\})$ is a child of D^* with respect to flipping u and v, then for each $x \in Z_{uv}$, $x \notin P_D[x]$.*

Proof. Recall that D^* is constructed from $D' = (D \setminus \{u\}) \cup X_{uv} \cup \{v\}$ by greedy removal of vertices. The greedy removal does not remove a vertex that is not dominated by other vertices, hence it will never place into Z_{uv} a vertex that is private for itself with respect to D' (and hence D). Furthermore, if $v_i \in Z_{uv}$, then when greedy removal of vertices considers a vertex v_j with $j > i$, if v_j is the unique vertex that dominates v_i with respect to the current dominating set at that point, it will never place v_j into Z_{uv}. $\square$

We are now ready to state our main theorem.

Theorem 3. *All minimal dominating sets of a chordal bipartite graph can be enumerated in incremental polynomial time. On input graphs with n vertices and m edges, the delay is $O(n^3 m|\mathcal{L}|^2)$, and the total running time is $O(n^3 m|\mathcal{L}^*|^2)$, where $\mathcal{L}$ is the family of already generated minimal dominating sets and $\mathcal{L}^*$ is the family of all minimal dominating sets.*

Proof. Let $G = (V_1, V_2, E)$ be a chordal bipartite graph. Let D^* be a minimal dominating set of G such that v is an isolated vertex of $G[D^*]$, and let u be a neighbor of v in G. We construct an algorithm that generates with delay $O(n^3)$ a family of minimal dominating sets $\mathcal{D}$ such that $\mathcal{D}$ contains all minimal dominating sets D that are children of D^* with respect to flipping u and v. By Lemma 1, this is sufficient to prove the theorem.

First of all, we output $\mathcal{D} = \emptyset$ and stop if there is $x \in N_G(v)$ such that $N_G(x) = \{v\}$, i.e., x is a pendant vertex adjacent to v. Assume from now on that this is not the case.

Let $R \subseteq (N_G(u) \setminus \{v\}) \cap D^*$ be the set of all vertices $x \in (N_G(u) \setminus \{v\}) \cap D^*$ such that $x \in P_{D^*}[x]$, i.e., these vertices are not dominated by other vertices of D^*. Denote by B the set of all vertices $y \in N_G(R \cup \{v\}) \setminus \{u\}$ such that $N_G(y) \cap D^* \subseteq R \cup \{v\}$. Observe that by the definition of R, $B \cap D^* = \emptyset$. Notice also that R and B are subsets of distinct sets of the bipartition of G. Without loss of generality we assume that $R \subseteq V_1$ and $B \subseteq V_2$. Let $R' = N_G(B) \setminus \{v\}$. Clearly, $R \subseteq R' \subseteq V_1$. Consider the red-blue bipartite graph $F = G[R' \cup B]$, where R' and B are the sets of red and blue vertices respectively. Using Theorem 2 we enumerate all minimal red dominating sets of F.

For each minimal red dominating set X of F, we consider the (not necessarily minimal) dominating set $D' = (D^* \setminus (R \cup \{v\})) \cup (X \cup \{u\})$ of G. We apply greedy removal of vertices to obtain a minimal dominating set D from D', and output D.

Denote by $\mathcal{D}$ the collection of generated sets. We prove the following two claims to show that $\mathcal{D}$ is a family of minimal dominating sets of G and $\mathcal{D}$ includes all children of D^* with respect to flipping u and v.

Claim A. All elements of $\mathcal{D}$ are pairwise distinct minimal dominating sets of G.

Proof of Claim A. Assume that $\mathcal{D} \neq \emptyset$. Let $Y = D^* \setminus (R \cup \{v\})$. Because D^* is a minimal dominating set and Y is a proper subset of D^*, Y is not a dominating set. Let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ be the sets of vertices that are not dominated by Y. Because all the vertices that are deleted from D^* to obtain Y are in V_1, $U_1 \subseteq R \cup \{v\}$. All these vertices are dominated by $u \in D'$, i.e., these vertices are dominated by D' and, consequently, by D. Each vertex $y \in U_2$ is dominated by D^* but is not dominated by Y. Hence, $y \in N(R \cup \{v\})$ and $N_G(y) \cap D^* \subseteq R \cup \{v\}$. Therefore, $B \subseteq U_2 \subseteq B \cup \{u\}$. Clearly, u is dominated by itself in D'. If there is a vertex $y \in B$ that is an isolated blue vertex of F, then y is adjacent only to v in G, but $\mathcal{D} = \emptyset$ in this case. We have that F has minimal red dominating sets, and each minimal red dominating set X of F dominates B, and $X \cup \{u\}$ dominates U_2. It follows that $D' = (D^* \setminus (R \cup \{v\})) \cup (X \cup \{u\})$ is a dominating set of G for each minimal red dominating set X of F. Then greedy removal of vertices produces a minimal dominating set produces the minimal dominating set D when applied to D'. We conclude that all elements of $\mathcal{D}$ are minimal dominating sets of G.

To show that the elements of $\mathcal{D}$ are pairwise distinct, consider D' and observe that because X is a minimal red dominating set of F, each vertex $x \in X$ has a private neighbor in B with respect to X in F and this vertex is not dominated by u as $u \in V_2$. Then these privates are privates with respect to D' in G. Hence, the vertices of X cannot be deleted by greedy removal of vertices. Since elements of $\mathcal{D}$ are constructed for distinct minimal red dominating sets X of F, the elements of $\mathcal{D}$ are distinct as well. $\square$

Claim B. For each child D of D^* with respect to flipping u and v, $D \in \mathcal{D}$.

Proof of Claim B. Let $D = (D^* \setminus (X_{uv} \cup \{v\})) \cup (Z_{uv} \cup \{u\})$ be a child of D^* with respect to flipping u and v; X_{uv} and Z_{uv} are the corresponding sets introduced in the definition of parents.

Suppose that v has a pendant neighbor x. Because $v \notin D$, $x \in D$. Since $v \in P_D(u)$, we have that $x = u$, but then u has no neighbors in $G[D]$. It contradicts the condition that u is not an isolated vertex of $G[D]$. We conclude that v has no pendant neighbors, and the algorithm does not stop when it checks the existence of such neighbors of v.

Observe that $X_{uv} \subseteq V_1$ and $Z_{uv} \subseteq V_1$. The vertices of X_{uv} are privates for u with respect to D, and for any $x \in X_{uv}$, $x \in P_{D^*}[x]$. Hence, $X_{uv} \subseteq R$. Let $z \in Z_{uv}$. We have that z has no private with respect to the dominating set $D'' = (D \setminus \{u\}) \cup X_{uv} \cup \{v\}$, but z has a private y with respect to D. It follows that $y \in N_G(X_{uv} \cup \{v\}) \setminus \{u\} \subseteq N_G(R) \setminus \{u\}$. Clearly, $y \notin D^*$, as the vertices of X_{uv} and v are isolated in D^*. Also any neighbor of y in D^* is in $X_{uv} \cup \{v\}$, because y is not dominated by $D^* \setminus (X_{uv} \cup \{v\})$. Therefore, $N_G(y) \cap D^* \subseteq X_{uv} \cup \{v\} \subseteq R \cup \{v\}$. Recall that B is the set of all vertices $y \in N_G(R \cup \{v\}) \setminus \{u\}$ such that $N_G(y) \cap D^* \subseteq R \cup \{v\}$. It follows that $y \in B$ and, therefore, $z \in R'$. Since this holds for every $z \in Z_{uv}$, we conclude that $Z_{uv} \subseteq R'$.

Next we show that $X = (R \setminus X_{uv}) \cup Z_{uv} \subseteq R'$ is a minimal red dominating set of F. Recall that $B \subseteq V_2 \setminus \{u\}$ and $B \cap D^* = \emptyset$. Because $Z_{uv} \subseteq V_1$, $B \cap D = \emptyset$. Let $y \in B$. By the definition, $N_G(y) \cap D^* \subseteq R \cup \{v\}$. Since D is a dominating set, y is dominated by a vertex of D, and if it is not dominated by $R \setminus X_{uv}$, it should be dominated by Z_{uv}. We conclude that X is a red dominating set of F.

To show minimality, we prove that for each $x \in X$, x has a private with respect to X in F. Suppose that $x \in R \setminus X_{uv}$. Since $x \in D$, $P_D[x] \neq \emptyset$. Furthermore, since $R \subseteq N_G(u)$, $P_D[x] = P_D(x)$. Let $y \in P_D(x)$. Then $y \neq u$ and $N_G(y) \cap D = \{x\}$, and we have that $N_G(y) \cap D^* \subseteq \{x\} \cup X_{uv} \cup \{v\} \subseteq R \cup \{v\}$, i.e., $y \in B$. Then y is a private for x with respect to X in F. Let $x \in Z_{uv} \setminus (R \setminus X_{uv})$. Because $x \in D$, $P_D[x] \neq \emptyset$. Since D is a child of D^*, D^* is a parent of D with respect to flipping u and v. By Observation 1, $P_D[x] = P_D(x)$. Let $y \in P_D(x)$. Vertex y is dominated in D^*, but it is not dominated by $D^* \setminus (X_{uv} \cup \{v\})$. Therefore, $y \in N_G(X_{uv} \cup \{v\}) \subseteq R \cup \{v\}$. Since $y \neq u$, $N_G(y) \cap D = \{x\}$ and $x \notin D^*$, we have that $N_G(y) \cap D^* \subseteq X_{uv} \cup \{v\} \subseteq R \cup \{v\}$, i.e., $y \in B$. Then y is a private for x with respect to X in F.

We proved that X is a minimal red dominating set of F. Therefore, X is generated by our algorithm. It remains to observe that $D = (D^* \setminus (R \cup \{v\})) \cup (X \cup \{u\})$. Since D is a minimal dominating set, then it is output, i.e, $D \in \mathcal{D}$. $\square$

To complete the proof, it remains to evaluate the running time. Observe that checking whether v has pendant neighbors can be done in time $O(n)$. Then the sets B, R, R' and the auxiliary graph F can be constructed in time $O(n^2)$. The minimal red dominating sets X of F can be enumerated with delay $O(n^3)$ by Theorem 2. For each generated set X, the minimal dominating set D is constructed in time $O(n^2)$. It follows that we generate elements of $\mathcal{D}$ with delay $O(n^3)$. $\square$

References

[1] D. Avis and K. Fukuda, *Reverse search for enumeration*, Discrete Applied Mathematics, 65 (1996), pp. 21–46.

[2] E. Boros, K. Elbassioni, and V. Gurvich, *Transversal hypergraphs to perfect matchings in bipartite graphs: characterization and generation algorithms*, Journal of Graph Theory, 53 (2006), pp. 209–232.

[3] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, *Generating Maximal Independent Sets for Hypergraphs with Bounded Edge-Intersections*, Proceedings of LATIN 2004, LNCS 2976, pp. 488–498 (2004).

[4] E. Boros, V. Gurvich, and P. L. Hammer, *Dual subimplicants of positive boolean functions*, Optimization Methods & Software, 10 (1998), pp. 147–156.

[5] E. Boros, P. L. Hammer, T. Ibaraki, and K. Kawakami, *Polynomial time recognition of 2-monotonic positive Boolean functions given by an oracle*, SIAM Journal on Computing, 26 (1997), pp. 93–109.

[6] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph classes: a survey*, SIAM (1999).

[7] B. Courcelle, *Linear delay enumeration and monadic second-order logic*, Discrete Applied Mathematics, 157 (2009), pp. 2675–2700.

[8] C. Domingo, N. Mishra, and L. Pitt, *Efficient read-restricted monotone cnf/dnf dualization by learning with membership queries*, Machine Learning, 37 (1999), pp. 89–110.

[9] T. Eiter, *Exact transversal hypergraphs and application to Boolean μ-functions*, Journal of Symbolic Computing, 17 (1994), pp. 215–225.

[10] T. Eiter and G. Gottlob, *Identifying the minimal transversals of a hypergraph and related problems*, SIAM Journal on Computing, 24 (1995), pp. 1278–1304.

[11] T. Eiter and G. Gottlob, *Hypergraph transversal computation and related problems in Logic and AI*, Proceedings of JELIA 2002, LNCS 2424, pp. 549–564 (2002).

[12] T. Eiter, G. Gottlob, and K. Makino, *New results on monotone dualization and generating hypergraph transversals*, SIAM Journal on Computing, 32 (2003), pp. 514–537. (Preliminary version in STOC 2002.)

[13] K. Elbassioni, K. Makino, and I. Rauf, *Output-sensitive algorithms for enumerating minimal transversals for some geometric hypergraphs*, Proceedings of ESA 2009, LNCS 5757, pp. 143–154 (2009).

[14] M. L. Fredman and L. Khachiyan, *On the complexity of dualization of monotone disjunctive normal forms*, Journal of Algorithms, 21 (1996), pp. 618–628.

[15] P. A. Golovach, P. Heggernes, D. Kratsch, and Y. Villanger, *An incremental polynomial time algorithm to enumerate all minimal edge dominating sets*, Algorithmica, to appear. DOI 10.1007/s00453-014-9875-7.

[16] M. C. Golumbic and C. F. Goss, *Perfect elimination and chordal bipartite graphs*, J. Graph Theory, 2 (1978), pp. 155–163.

[17] T. W. Haynes and S. T. Hedetniemi, *Domination in graphs*, Marcel Dekker Inc., New York (1998).

[18] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, *On generating all maximal independent sets*, Information Processing Letters, 27 (1988), pp. 119–123.

[19] M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine, *On the enumeration of minimal dominating sets and related notions.* SIAM J. Discrete Math., 28:4 (2014), pp. 1916-1929. (Preliminary version in FCT 2011.)

[20] M. M. Kanté, V. Limouzy, A. Mary, and 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).

[21] M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine, *On the Enumeration and Counting of Minimal Dominating sets in Interval and Permutation Graphs*, Proceedings of ISAAC 2013, LNCS 8283, pp. 339-349 (2013).

[22] M. M. Kanté, V. Limouzy, A. Mary, L. Nourine and T. Uno, *A Polynomial Delay Algorithm for Enumerating Minimal Dominating Sets in Chordal Graphs*, arxiv:1407.2036 (2014).

[23] M. M. Kanté, V. Limouzy, A. Mary, L. Nourine and T. Uno, *Polynomial Delay Algorithm for Listing Minimal Edge Dominating sets in Graphs*, arxiv:1404.3501 (2014).

[24] L. Khachiyan, E. Boros, K. Borys, K. M. Elbassioni, and V. Gurvich, *Generating all vertices of a polyhedron is hard*, Discrete & Computational Geometry, 39 (2008), pp. 174–190.

[25] L. Khachiyan, L. Boros, K. Borys, K. M. Elbassioni, V. Gurvich, and K. Makino, *Generating Cut Conjunctions in Graphs and Related Problems*, Algorithmica, 51 (2008), pp. 239–263.

[26] L. Khachiyan, E. Boros, K. M. Elbassioni, and V. Gurvich, *On enumerating minimal dicuts and strongly connected subgraphs*, Algorithmica, 50 (2008), pp. 159–172.

[27] T. Kloks and D. Kratsch, *Computing a perfect edge without vertex elimination ordering of a chordal bipartite graph*, Inf. Process. Lett., 55 (1995), pp. 11–16.

[28] E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Generating all maximal independent sets: NP-hardness and polynomial-time algorithms*, SIAM Journal on Computing, 9 (1980), pp. 558–565.

[29] K. Makino and T. Ibaraki, *The maximum latency and identification of positive Boolean functions*, SIAM J. Comput., 26 (1997), pp. 1363-1383.

[30] K. Makino and T. Ibaraki, *A fast and simple algorithm for identifying 2-monotonic positive Boolean functions*, Journal of Algorithms, 26 (1998), pp. 293–305.

[31] C. Papadimitriou, *NP-completeness: A retrospective*, Proceedings of ICALP 1997, LNCS 1256, pp. 2–6 (1997).

[32] B. Schwikowski and E. Speckenmeyer, *On enumerating all minimal solutions of feedback problems*, Discrete Applied Mathematics, 117 (2002), pp. 253–265.

[33] R. E. Tarjan, *Enumeration of the elementary circuits of a directed graph*, SIAM Journal on Computing, 2 (1973), pp. 211–216.

[34] S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi, *An algorithm to enumerate all cutsets of a graph in linear time per cutset*, Journal of the ACM, 27 (1980), pp. 619–632.