An incremental polynomial time algorithm to enumerate all minimal edge dominating sets*

Petr A. Golovach[1] Pinar Heggernes[1] Dieter Kratsch[2] Yngve Villanger[1]

[1] Department of Informatics, University of Bergen, Norway,
{petr.golovach,pinar.heggernes,yngve.villanger}@ii.uib.no
[2] LITA, Université de Lorraine - Metz, France,
kratsch@univ-metz.fr

Abstract. We show that all minimal edge dominating sets of a graph can be generated in incremental polynomial time. We present an algorithm that solves the equivalent problem of enumerating minimal (vertex) dominating sets of line graphs in incremental polynomial, and consequently output polynomial, time. Enumeration of minimal dominating sets in graphs has very recently been shown to be equivalent to enumeration of minimal transversals in hypergraphs. The question whether the minimal transversals of a hypergraph can be enumerated in output polynomial time is a fundamental and challenging question; it has been open for several decades and has triggered extensive research. To obtain our result, we present a flipping method to generate all minimal dominating sets of a graph. Its basic idea is to apply a flipping operation to a minimal dominating set D^* to generate minimal dominating sets D such that $G[D]$ contains more edges than $G[D^*]$. We show that the flipping method works efficiently on line graphs, resulting in an algorithm with delay $O(n^2m^2|\mathcal{L}|)$ between each pair of consecutively output minimal dominating sets, where n and m are the numbers of vertices and edges of the input graph, respectively, and $\mathcal{L}$ is the set of already generated minimal dominating sets. Furthermore, we are able to improve the delay to $O(n^2m|\mathcal{L}|)$ on line graphs of bipartite graphs. Finally we show that the flipping method is also efficient on graphs of large girth, resulting in an incremental polynomial time algorithm to enumerate the minimal dominating sets of graphs of girth at least 7.

1 Introduction

Enumerating, i.e., generating or listing, all vertex or edge subsets of a graph that satisfy a specified property plays a central role in graph algorithms; see e.g. [1, 2, 8–10, 16, 20–22, 24, 29, 30, 32]. Enumeration algorithms with running time that is polynomial in the size of the input plus the size of the output are called output

* This work is supported by the European Research Council, the Research Council of Norway, and the French National Research Agency. Due the space restrictions, proofs of various claims are omitted. The complete version of the paper can be found on-line in [13].

polynomial time algorithms. For various enumeration problems it has been shown that no output polynomial time algorithm can exist unless P = NP [20, 22, 24]. A potentially better behavior than output polynomial time is achieved by so called incremental polynomial time algorithms, which means that the next set in the list of output sets is generated in time that is polynomial in the size of the input plus the size of the already generated part of the output. Incremental polynomial time immediately implies output polynomial time.

One of the most classical and widely studied enumeration problems is that of listing all minimal transversals of a hypergraph, i.e., minimal hitting sets of its set of hyperedges. This problem has applications in areas like database theory, machine learning, data mining, game theory, artificial intelligence, mathematical programming, and distributed systems; extensive lists of corresponding references are provided by e.g., Eiter and Gottlob [10], and Elbassioni, Makino, and Rauf [11]. Whether or not all minimal transversals of a hypergraph can be listed in output polynomial time has been identified as a fundamental challenge in a long list of seminal papers, e.g., [8–12, 16, 27], and it remains unresolved despite continuous attempts since the 1980's.

Recently Kanté, Limouzy, Mary, and Nourine [18] have proved that enumerating the minimal transversals of a hypergraph is equivalent to enumerating the minimal dominating sets of a graph. In particular, they show that an output polynomial time algorithm for enumerating minimal dominating sets in graphs implies an output polynomial time algorithm for enumerating minimal transversals in hypergraphs. Dominating sets form one of the best studied notions in computer science; the number of papers on domination in graphs is in the thousands, and several well known surveys and books are dedicated to the topic (see, e.g., [14]).

Given the importance of the hypergraph transversal enumeration problem and the failed attempts to resolve whether it can be solved in output polynomial time, efforts to identify tractable special cases have been highly appreciated [3, 4, 6–8, 10, 11, 25, 26]. The newly proved equivalence to domination allows for new ways to attack this long-standing open problem. In fact some results on output polynomial algorithms to enumerate minimal dominating sets in graphs already exist for graphs of bounded treewidth and of bounded clique-width [5], interval graphs [8], strongly chordal graphs [8], planar graphs [10], degenerate graphs [10], and split graphs [17].

In this paper we show that all minimal dominating sets of line graphs and of graphs of large girth can be enumerated in incremental polynomial time. More precisely, we give algorithms where the time delay between two consecutively generated minimal dominating sets is $O(n^2m^2|\mathcal{L}|)$ on line graphs, $O(n^2m|\mathcal{L}|)$ on line graphs of bipartite graphs, and $O(n^2m|\mathcal{L}|^2)$ on graphs of girth at least 7, where $\mathcal{L}$ is the set of already generated minimal dominating sets of an input graph on n vertices and m edges. Line graphs form one of the oldest and most studied graph classes [15, 23, 33] and they can be recognized in linear time [28]. Our results, in addition to proving tractability for two substantial cases of the hypergraph transversal enumeration problem, imply incremental polynomial time enumeration of minimal edge dominating sets in *arbitrary* graphs. In particular, we obtain an algorithm with delay $O(m^6|\mathcal{L}|)$ to enumerate all minimal edge dominating sets of any graph on m edges, where $\mathcal{L}$ is the set of already generated edge dominating sets. For bipartite graphs, we are able to reduce the delay to $O(m^4|\mathcal{L}|)$.

Our algorithms are based on the supergraph technique for enumerating vertex subsets in graphs [2, 21, 29, 32]. As a central tool in our algorithms, we present a new *flipping* method to generate the out-neighbors of a node of the supergraph, in other words, to generate new minimal dominating sets from a parent dominating set. Given a minimal dominating set D^*, our flipping operation replaces an isolated vertex of $G[D^*]$ with a neighbor outside of D^*, and, if necessary, supplies the resulting set with additional vertices to obtain new minimal dominating sets D, such that $G[D]$ has more edges compared to $G[D^*]$. Each of our algorithms starts with enumerating all maximal independent sets of the input graph G using the algorithm of Johnson, Papadimitriou, and Yannakakis [16], 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. We show that on all graphs, the flipping method enables us to identify a unique parent for each minimal dominating set. On line graphs and graphs of girth at least 7, we are able to prove additional (different) properties of the parents, which allow us to obtain the desired running time on these graph classes.

In a very recent publication of their work that was simultaneous with and independent from our work, Kanté, Limouzy, Mary, and Nourine [19] give output polynomial time algorithms for enumerating the minimal dominating sets of line graphs and path graphs. Their method is completely different from ours, as they obtain their algorithms through proving that line graphs and path graphs have closed neighborhood hypergraphs of bounded conformality.

2 Definitions and Preliminary Results

As input graphs to our enumeration problem, we consider finite undirected graphs without loops or multiple edges. Given such a graph $G = (V, E)$, its vertex and edge sets, V and E, are also denoted by $V(G)$ and $E(G)$, respectively. The subgraph of G induced by a subset $U \subseteq V$ is denoted by $G[U]$. For a vertex v, we denote by $N(v)$ its *(open) neighborhood*, that is, the set of vertices that are adjacent to v. The *closed neighborhood* of v is the set $N(v) \cup \{v\}$, and it is denoted by $N[v]$. If $N(v) = \emptyset$ then v is *isolated*. For a set $U \subseteq V$, $N[U] = \cup_{v \in U} N[v]$, and $N(U) = N[U] \setminus U$. The *girth* $g(G)$ of a graph G is the length of a shortest cycle in G; if G has no cycles, then $g(G) = +\infty$. A set of vertices is a *clique* if it induces a complete subgraph of G. A clique is *maximal* if no proper superset of it is a clique.

Two edges in E are adjacent if they share an endpoint. The *line graph $L(G)$ of G* is the graph whose set of vertices is $E(G)$, such that two vertices e and e' of $L(G)$ are adjacent if and only if e and e' are adjacent edges of G. A graph H is a *line graph* if H is isomorphic to $L(G)$ for some graph G. Equivalently, a graph is a line graph if its edges can be partitioned into maximal cliques such that no vertex lies in more than two maximal cliques. This implies in particular that the neighborhood of every vertex can be partitioned into at most two cliques. It is well known that line graphs do not have induced subgraphs isomorphic to $K_{1,3}$, also called a *claw*.

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. A dominating set is *minimal* if no proper subset of it is a 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 vertex v (with respect to D) if u is dominated by v but is not nominated by $D \setminus \{v\}$. Clearly, 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. Notice that 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 $P_D[v] = P_D(v)$.

A set of edges $A \subseteq E$ is an *edge dominating* set if each edge $e \in E$ is either in A or is adjacent to an edge in A. An edge dominating set is *minimal* if no proper subset of it is an edge dominating set. It is easy to see that A is a (minimal) edge dominating set of G if and only if A is a (minimal) dominating set of $L(G)$.

Let $\phi(X)$ be a property of a set of vertices or edges X of a graph, e.g., "X is a minimal dominating set". The *enumeration problem for property* $\phi(X)$ for a given graph G on n vertices and m edges asks for the set $\mathcal{C}$ of all subsets of vertices or edges X of G that satisfy $\phi(X)$. An *enumeration algorithm* is an algorithm that solves this problem, i.e., that lists the elements of $\mathcal{C}$ without repetitions. An enumeration algorithm $\mathcal{A}$ is said to be *output polynomial time* if there is a polynomial $p(x, y)$ such that all elements of $\mathcal{C}$ are listed in time bounded by $p((n+m), |\mathcal{C}|)$. Assume now that $X_1, \ldots, X_\ell$ are the elements of $\mathcal{C}$ enumerated in the order in which they are generated by $\mathcal{A}$. The *delay* of $\mathcal{A}$ is the maximum time $\mathcal{A}$ requires between outputting X_{i-1} and X_i, for $i \in \{1, \ldots, \ell\}$. Algorithm $\mathcal{A}$ is *incremental polynomial time* if there is a polynomial $p(x, i)$ such that for each $i \in \{1, \ldots, \ell\}$, X_i is generated in time bounded by $p((n+m), i)$. Finally, $\mathcal{A}$ is a *polynomial delay* algorithm if there is a polynomial $p(x)$ such that for each $i \in \{1, \ldots, \ell\}$, the delay between outputting X_{i-1} and X_i is at most $p(n+m)$.

A set of vertices $U \subseteq V$ is an *independent set* if no two vertices of U are adjacent in G, and an independent set is *maximal* if no proper superset of it is an independent set. The following observation is folklore.

Observation 1. *Every maximal independent set of a graph G is a minimal dominating set of G. Furthermore, the set of all maximal independent sets of G is exactly the set of all its minimal dominating sets D such that $G[D]$ has no edges.*

Theorem 1 ([16]). *All maximal independent sets of a graph with n vertices and m edges can be enumerated in lexicographic order with polynomial delay $O(n(m+n\log|\mathcal{I}|))$, where $\mathcal{I}$ is the set of already generated maximal independent sets.*

Let $v_1, \ldots, v_n$ be the vertices of a graph G. Suppose that D' is a dominating set of G. We say that a 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.

Finally, give some definitions on directed graphs, as the supergraph technique that we use creates an auxiliary directed graph. To distinguish this graph from the input graph, we will call the vertices of a directed graph *nodes*. The edges of a directed graph have directions and are called *arcs*. An arc (u, v) has direction from node u to node v. The *out-neighbors* of a node u are all nodes v such that (u, v) is an arc. Similarly, the *in-neighbors* of a node v are all nodes u such that (u, v) is an arc. In this paper, an in-neighbor will sometimes be called a *parent*.

3 Enumeration by flipping: the general approach

In this section we describe the general scheme of our enumeration algorithms. Let G be a graph; we fix an (arbitrary) order of its vertices: $v_1, \ldots, v_n$. Observe that this order induces a lexicographic order on the set $2^{V(G)}$. Whenever greedy removal of vertices of a dominating set is performed further in the paper, 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 $X_{uv} \cup \{v\}$. Let Z_{uv} be the set of vertices that are removed to ensure minimality, and let $D^* = ((D \setminus \{u\}) \cup X_{uv} \cup \{v\}) \setminus Z_{uv}$.

Lemma 1. *The set D^* is a minimal dominating set in G such that $X_{uv} \cup \{v\} \subseteq D^*$, $|E(G[D^*])| < |E(G[D])|$ and v is an isolated vertex of $G[D^*]$.*

Our main tool, 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).

Assume that there is an enumeration algorithm $\mathcal{A}$ that, given a minimal dominating set D^* of a graph G such that $G[D^*]$ has isolated vertices, an isolated vertex v of $G[D^*]$, and a neighbor u of v in G, generates with polynomial delay a set 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. In this case we can enumerate all minimal dominating sets of the graph G with n vertices and m edges as follows.

Our method is a variant of the supergraph technique that has been applied for enumerating subsets with various properties in graphs [2, 21, 29, 32]. More precisely, we define a directed graph $\mathcal{G}$ whose nodes are minimal dominating sets of G, with an additional special node r, called the *root*, that has no in-neighbors. Recall that by Observation 1., maximal independent sets are minimal dominating sets, i.e., they are nodes of $\mathcal{G}$. We add an arc from the root r to every maximal independent set of G. For each minimal dominating set $D^* \in V(\mathcal{G})$, we add an arc from D^* to every minimal dominating set D such that $\mathcal{A}$ generates D from D^* for some choice of u and v.

Next we run Depth-First Search in $\mathcal{G}$ starting from r. Observe that we need not construct $\mathcal{G}$ explicitly to do this, as for each node $W \neq r$ of $\mathcal{G}$ we can use $\mathcal{A}$ to generate all out-neighbors of W, and we can generate the out-neighbors of r with polynomial delay by Theorem 1. Hence, we maintain a list $\mathcal{L}$ of minimal dominating sets of G sorted in lexicographic order that are already visited nodes of $\mathcal{G}$. Also we keep a stack $\mathcal{S}$ of records R_W for $W \in V(\mathcal{G})$ that are on the path from r to the current node of $\mathcal{G}$. These records are used to generate out-neighbors. The record R_r contains the last generated maximal independent set and the information that is necessary to proceed with the enumeration of maximal independent sets. Each of the records R_W, for $W \neq r$, contains the current choice of u and v, the last set D generated by $\mathcal{A}$ for the instance (W, u, v), and the information that is necessary for $\mathcal{A}$ to proceed with the enumeration.

Lemma 2. *Suppose that $\mathcal{A}$ generates the elements of $\mathcal{D}$ for a triple (D^*, u, v) with polynomial delay $O(p(n, m))$. Let $\mathcal{L}^*$ be the set of all minimal dominating sets. Then the algorithm described above enumerates all minimal dominating sets as follows:*

- *with delay $O((p(n, m) + n^2)m|\mathcal{L}|^2)$ and total running time $O((p(n, m) + n^2)m|\mathcal{L}^*|^2)$;*
- *if $|E(G[D])| > |E(G[D^*])|$ for every $D \in \mathcal{D}$, then the delay is $O((p(n, m) + n^2)m^2|\mathcal{L}|)$, and the total running time is $O((p(n, m) + n^2)m|\mathcal{L}^*|^2)$;*
- *if $\mathcal{D}$ contains only children of D^* with respect to flipping of u and v, then the delay is $O((p(n, m) + n^2)m|\mathcal{L}|)$, and the total running time is $O((p(n, m) + n^2)m|\mathcal{L}^*|)$.*

Proof. Recall that any minimal dominating set D with at least one edge has a parent D^* and $|E(G[D^*])| < |E(G[D])|$. Because $\mathcal{A}$ generates D from D^*, (D^*, D) is an arc in $\mathcal{G}$. It follows that for any minimal dominating set $D \in V(\mathcal{G})$ with at least one edge, there is a maximal independent set $I \in V(\mathcal{G})$ such that I and D are connected by a directed path in $\mathcal{G}$. As (r, I) is an arc in $\mathcal{G}$, D is reachable from r. We conclude that Depth-First Search visits, and thus enumerates all nodes of $\mathcal{G}$. It remains to evaluate the running time.

To get a new minimal dominating set, we consider the records in $\mathcal{S}$. For each record R_W for $W \neq r$, we have at most m possibilities for u and v to get a new set D. As soon as a new set is generated it is added to $\mathcal{L}$ unless it is already in $\mathcal{L}$. Hence, we generate at most $m|\mathcal{L}|$ sets for W in time $(p(n,m) + n^2)m|\mathcal{L}|$, as each set is generated with polynomial delay $O(p(n,m))$, and after its generation we immediately test whether or not it is already in $\mathcal{L}$, which takes $O(n \log |\mathcal{L}|) = O(n^2)$ time, because $|\mathcal{L}| \leq 2^n$. For R_r, we generate at most $|\mathcal{L}|$ sets. Because any isolated vertex of G belongs to every maximal independent set, each set is generated with delay $O(n'(m + n' \log |\mathcal{L}|))$, i.e., in time $O(n'(m + n'^2))$ by Theorem 1, where n' is the number of non-isolated vertices. As $n' \leq 2m$, these sets are generated in time $O(n^2 m|\mathcal{L}|)$. Since $|\mathcal{S}| \leq |\mathcal{L}|$, in time $O((p(n,m) + n^2)m|\mathcal{L}|^2)$ we either obtain a new minimal dominating set or conclude that the list of minimal dominating sets is exhausted.

To get the bound for the total running time, recall that Depth-First Search runs in time that is linear in $|E(\mathcal{G})|$. As for each arc we perform $O((p(n,m) + n^2)m)$ operations, the total running time is $O((p(n,m) + n^2)m|\mathcal{L}^*|^2)$.

If for every $D \in \mathcal{D}$, $|E(G[D])| > |E(G[D^*])|$, then the delay is less. To see this, we observe that the number of edges in any minimal dominating set is at most m. Hence, any directed path starting from r in $\mathcal{G}$ has length at most m and, therefore, $|\mathcal{S}| \leq m + 1$. By the same arguments as above, we get that in time $O((p(n,m) + n^2)m^2|\mathcal{L}|)$ we either obtain a new minimal dominating set or conclude that the list of minimal dominating sets is complete.

Assume finally that $\mathcal{D}$ contains only children of D^* with respect to flipping of u and v. Since each minimal dominating set D with $E(G[D]) \neq \emptyset$ has a unique parent with respect to flipping of any vertices $u \in D \cap N[D \setminus \{u\}]$ and $v \in P_D(u)$, each D has at most m parents. Hence, we generate at most $m|\mathcal{L}|$ sets until we obtain a new minimal dominating set or conclude that the list is exhausted. As to generate a set and check whether it is already listed we spend time $O(p(n,m) + n^2)$, the delay between two consecutive sets that are output is $O((p(n,m)+n^2)m|\mathcal{L}|)$ and the total running time is $O((p(n,m)+n^2)m|\mathcal{L}^*|)$. □

To be able to apply our method, we have to show how to construct an algorithm, like algorithm $\mathcal{A}$ above, that produces $\mathcal{D}$ with polynomial delay. We will use the following lemma for this purpose.

Lemma 3. *Let D be a child of D^* with respect to flipping u and v; $D^* = ((D \setminus \{u\}) \cup X_{uv} \cup \{v\}) \setminus Z_{uv}$. Then for every vertex $z \in Z_{uv}$, the following three statements are true:*

1. $z \notin N[X_{uv} \cup \{v\}]$,

2. *z is dominated by a vertex of $D^* \setminus (X_{uv} \cup \{v\})$,*
3. *there is a vertex $x \in N[X_{uv} \cup \{v\}] \setminus N[u]$ adjacent to z such that $x \notin N[D^* \setminus (X_{uv} \cup \{v\})]$.*

Furthermore, for every $x \in N[X_{uv} \cup \{v\}] \setminus N[u]$ such that $x \notin N[D^* \setminus (X_{uv} \cup \{v\})]$, there is a vertex $z \in Z_{uv}$ such that x and z are adjacent.

We use this lemma to construct an algorithm for generating $\mathcal{D}$. The idea is to generate $\mathcal{D}$ by considering all possible candidates for X_{uv} and Z_{uv}. It would be interesting to know whether this can be done efficiently in general. On line graphs and graphs of girth at least 7, we are able to prove additional properties of the parent minimal dominating sets which result in efficient algorithms for generating $\mathcal{D}$, as will be explained in the sections below.

4 Enumeration of minimal edge dominating sets

In this section we show that all minimal edge dominating sets of an *arbitrary* graph can be enumerated in incremental polynomial time. We achieve this by enumerating the minimal dominating sets in line graphs.

For line graphs, we construct an enumeration algorithm that, given a minimal dominating set D^* of a graph G such that $G[D^*]$ has isolated vertices, an isolated vertex v of $G[D^*]$, and a neighbor u of v in G, generates with polynomial delay a set of minimal dominating sets $\mathcal{D}$ that contains all children of D^* with respect to flipping u and v, and has the property that $|E(G[D])| > |E(G[D^*])|$, for every $D \in \mathcal{D}$.

This is possible because on line graphs we can prove additional properties of a parent in the flipping method. Let D be a minimal dominating set of a graph G such that $G[D]$ has at least one edge uw, and assume that $v \in P_D(u)$. Recall that D^* is defined by choosing a maximal independent set $X_{uv} \subseteq P_D(u) \setminus N[v]$ in $G[P_D(u) \setminus N[v]]$, then considering the set $D' = (D \setminus \{u\}) \cup X_{uv} \cup \{v\}$, and letting $D^* = D' \setminus Z_{uv}$ where $Z_{uv} \subseteq D \cap D'$.

Lemma 4. *If G is a line graph, then:*

- *$X_{uv} = \emptyset$,*
- *each vertex of Z_{uv} is adjacent to exactly one vertex of $P_{D^*}(v) \setminus N[u]$,*
- *each vertex of $P_{D^*}(v) \setminus N[u]$ is adjacent to exactly one vertex of Z_{uv}.*

Consider a line graph G with n vertices $v_1, \ldots, v_n$ and m edges. Let D^* be a minimal dominating set and let v be an isolated vertex of $G[D^*]$. Suppose that u is a neighbor of v. Let $\{x_1, \ldots, x_k\} = P_{D^*}(v) \setminus N[u]$. We construct minimal dominating sets from $(D^* \setminus \{v\}) \cup \{u\}$ by adding a set Z that contains a neighbor of each x_i from $N(x_i) \setminus N[v]$. Recall that the vertices $x_1, \ldots, x_k$ should be dominated by Z_{uv} for every child of D^* by Lemma 3, and by the same lemma each x_i is dominated by a vertex from $N(x_i) \setminus N[v]$.

Let $U = N[u] \cup (\bigcup_{i=1}^{k} (N[x_i] \setminus N[v]) \cup \{x_i\})$. We need the following observation that also will be used in the next section. To see its correctness, it is sufficient to notice that because G contains no claws, $N[x_i] \setminus N[v]$ is a clique.

Lemma 5. *For any choice of a set $Z = \{z_1, \ldots, z_k\}$ such that $z_i \in N(x_i) \setminus N[v]$ for $i \in \{1, \ldots, k\}$, U is dominated by $Z \cup \{u\}$.*

We want to ensure that by subsequent removal of vertices of $D^* \setminus \{v\}$ (which we do to guarantee minimality), the number of edges in the obtained minimal dominating set is not decreased. To do it, for each vertex $v_j \in V(G)$, we construct the sets of vertices R_j that cannot belong to Z_{uv} for any child D of D^*, where both D and D^* contain v_j. First, we set $R_j = \emptyset$ for every $v_j \notin D^* \setminus \{v\}$. Let v_j be a vertex of $D^* \setminus \{v\}$ that has a neighbor v_s such that either $v_s \in D^*$ or $v_s = u$. As G does not contain a claw, $K = N(v_j) \setminus N[v_s]$ is a clique. Then we set $R_j = K$ in this case. Notice that we can have several possibilities for v_s. In this case v_s is chosen arbitrary. For all other $v_j \in D^* \setminus \{v\}$, $R_j = \emptyset$. Denote by R the set $\cup_{j=1}^{n} R_j$. For each $i \in \{1, \ldots, k\}$, let

$$Z_i = \{z \in V | z \in N(D^* \setminus \{v\}) \cap (N(x_i) \setminus (N[v] \cup R)), N(z) \cap (P_{D^*}(v) \setminus N[u]) = \{x_i\}\}.$$

We generate a set $\mathcal{D}$ of minimal dominating sets as follows.

Case 1. If at least one of the following three conditions is fulfilled, then we set $\mathcal{D} = \emptyset$:

i) there is a vertex $x \in D^* \setminus \{v\}$ such that $N[x] \subseteq N[D^* \setminus \{v, x\}] \cup U$,
ii) $k \geq 1$ and there is an index $i \in \{1, \ldots, k\}$ such that $Z_i = \emptyset$,
iii) u is not adjacent to any vertex of $D^* \setminus \{v\}$ and $N(u) \cap (\cup_{i=1}^{k} Z_i) = \emptyset$.

Otherwise, we consider two other cases.

Case 2. If u is adjacent to a vertex of $D^* \setminus \{v\}$, then we consecutively construct all sets $Z = \{z_1, \ldots, z_k\}$ where $z_i \in Z_i$, for $1 \leq i \leq k$ (if $k = 0$, then $Z = \emptyset$). For each Z, we construct the set $D' = (D^* \setminus \{v\}) \cup \{u\} \cup Z$. Notice that D' is a dominating set as all vertices of $P_{D^*}[v]$ are dominated by D', but D' is not necessarily minimal. Hence, we construct a minimal dominating set D from D' by the greedy removal of vertices. The obtained set D is unique for a given set Z, and it is added to $\mathcal{D}$.

Recall that by the definition of the parent-child relation, u should be dominated by a vertex in a child. If u is not adjacent to a vertex of $D^* \setminus \{v\}$, it should be adjacent to at least one of the added vertices. This gives us the next case.

Case 3. If u is not adjacent to any vertex of $D^* \setminus \{v\}$, and $N(u) \cap (\bigcup_{i=1}^{k} Z_i) \neq \emptyset$, then we proceed as follows. Let j be the smallest index such that $N(u) \cap Z_j \neq \emptyset$, and let j' be the smallest index at least j such that $Z_{j'} \setminus N(u) = \emptyset$ ($j' = k$ if they are all non-empty). For each t starting from $t = j$ and continuing until $t = j'$, we do the following. If $N(u) \cap Z_t = \emptyset$ then we go to next step $t = t + 1$. Otherwise, for each $w \in N(u) \cap Z_t$, we consider all possible sets $Z = \{z_1, \ldots, z_{t-1}, z_{t+1}, \ldots, z_k\} \cup \{w\}$ such that $z_i \in Z_i \setminus N(u)$ for $1 \leq i \leq t - 1$, and $z_i \in Z_i$ for $t + 1 \leq i \leq k$. As above, for each such set Z, we construct the set $D' = (D^* \setminus \{v\}) \cup \{u\} \cup Z$ and then create a minimal dominating set D from D' by the greedy removal of vertices. The obtained set D is unique for a given set Z, and it is added to $\mathcal{D}$.

We summarize the properties of the above algorithm in the following lemma.

Lemma 6. *The set $\mathcal{D}$ is a set of minimal dominating sets such that $\mathcal{D}$ contains all children of D^* with respect to flipping u and v. Furthermore, $|E(G[D])| > |E(G[D^*])|$ for every $D \in \mathcal{D}$, and the elements of $\mathcal{D}$ are generated with delay $O(n+m)$.*

Combining Lemmas 2 and 6, we obtain the following theorem and corollary.

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

Corollary 1. *All minimal edge dominating sets of an arbitrary graph be enumerated in incremental polynomial time. On input graphs with m edges, the delay is $O(m^6|\mathcal{L}|)$ and the total running time is $O(m^4|\mathcal{L}^*|^2)$, where $\mathcal{L}$ is the set of already generated minimal edge dominating sets and $\mathcal{L}^*$ is the set of all minimal edge dominating sets.*

We can improve the dependence of the total running time on the size of the output if we restrict our attention to edge dominating sets of bipartite graphs. Again, we work on the equivalent problem of generating minimal dominating sets of line graphs of bipartite graphs.

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

Corollary 2. *All minimal edge dominating sets of a bipartite graph edges can be enumerated in incremental polynomial time. On input graphs with m edges, the delay is $O(m^4|\mathcal{L}|)$, and the total running time is $O(m^4|\mathcal{L}^*|)$, where $\mathcal{L}$ is the set of already generated minimal dominating sets, and $\mathcal{L}^*$ is the set of all minimal edge dominating sets.*

5 Graphs of large girth and concluding remarks

On line graphs we were able to observe properties of the parent relation in addition to uniqueness, which made it possible to apply the flipping method and design efficient algorithms for enumerating the minimal dominating sets. As another application of the flipping method, we show that it also works on graphs of girth at least 7. To do this, we observe other desirable properties of the parent relation on this graph class. As a result, we obtain an algorithm that enumerates the minimal dominating sets of a graph of girth at least 7 with delay $O(n^2m|\mathcal{L}|^2)$.

To conclude, the flipping method that we have described in this paper has the property that each generated minimal dominating set has a unique parent. It would be very interesting to know whether this can be used to obtain output polynomial time algorithms for enumerating minimal dominating sets in general. For the algorithms that we have given in this paper, on the studied graph classes we were able to give additional properties of the parents to obtain the desired running times. Are there additional properties of parents in general graphs that can result in efficient algorithms?

As a first step towards resolving these questions, on which other graph classes can the flipping method be used to enumerate the minimal dominating sets in output polynomial time? Another interesting question is whether the minimal dominating sets of line graphs or graphs of large girth can be enumerated with polynomial delay.

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, V. Gurvich, and P. L. Hammer, *Dual subimplicants of positive boolean functions*, Optimization Methods & Software, 10 (1998), pp. 147–156.
4. 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.
5. B. Courcelle, *Linear delay enumeration and monadic second-order logic*, Discrete Applied Mathematics, 157 (2009), pp. 2675–2700.
6. 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.
7. T. Eiter, *Exact transversal hypergraphs and application to Boolean μ-functions*, Journal of Symbolic Computing, 17 (1994), pp. 215–225.
8. T. Eiter and G. Gottlob, *Identifying the minimal transversals of a hypergraph and related problems*, SIAM Journal on Computing, 24 (1995), pp. 1278–1304.
9. 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).
10. 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.)
11. 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).
12. M. L. Fredman and L. Khachiyan, *On the complexity of dualization of monotone disjunctive normal forms*, Journal of Algorithms, 21 (1996), pp. 618–628.
13. P. A. Golovach, P. Heggernes, D. Kratsch, and Y. Villanger, *Generating all minimal edge dominating sets with incremental-polynomial delay*, CoRR, abs/1208.5345, 2012.

14. T. W. Haynes and S. T. Hedetniemi, *Domination in graphs*, Marcel Dekker Inc., New York (1998).
15. R. L. Hemminger and L. W. Beineke, *Line graphs and line digraphs*, in L. W. Beineke and R. J. Wilson (eds), Selected Topics in Graph Theory, Academic Press, pp. 271–305 (1978).
16. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, *On generating all maximal independent sets*, Information Processing Letters, 27 (1988), pp. 119–123.
17. M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine, *Enumeration of minimal dominating sets and variants*, Proceedings of FCT 2011, LNCS 6914, pp. 298–394 (2011).
18. M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine, *On the enumeration of minimal dominating sets and related notions*, submitted for journal publication, available at `http://www.isima.fr/~kante/research.php`.
19. 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).
20. 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.
21. 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.
22. L. Khachiyan, E. Boros, K. M. Elbassioni, and V. Gurvich, *On enumerating minimal dicuts and strongly connected subgraphs*, Algorithmica, 50 (2008), pp. 159–172.
23. J. Krausz, *Démonstration nouvelle d'un théorème de Whitney sur les réseaux*, Mat. Fiz. Lapok, 50 (1943), pp. 75–85.
24. 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.
25. K. Makino and T. Ibaraki, *The maximum latency and identification of positive Boolean functions*, SIAM J. Comput., 26 (1997), pp. 1363-1383.
26. 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.
27. C. Papadimitriou, *NP-completeness: A retrospective*, Proceedings of ICALP 1997, LNCS 1256, pp. 2–6 (1997).
28. N. D. Roussopoulos, *A max $\{m, n\}$ algorithm for determining the graph H from its line graph G*, Information Processing Letters, 2 (1973), pp. 108–112.
29. B. Schwikowski and E. Speckenmeyer, *On enumerating all minimal solutions of feedback problems*, Discrete Applied Mathematics, 117 (2002), pp. 253–265.
30. R. E. Tarjan, *Enumeration of the elementary circuits of a directed graph*, SIAM Journal on Computing, 2 (1973), pp. 211–216.
31. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, *A new algorithm for generating all the maximal independent sets*, SIAM Journal on Computing, 6 (1977), pp. 505–517.
32. 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.
33. H. Whitney, *Congruent graphs and the connectivity of graphs*, American Journal of Mathematics, 54 (1932), pp. 150–168.