ON THE ENUMERATION OF MINIMAL DOMINATING SETS AND RELATED NOTIONS

MAMADOU MOUSTAPHA KANTÉ, VINCENT LIMOUZY, ARNAUD MARY, AND LHOUARI NOURINE

Abstract. A dominating set D in a graph is a subset of its vertex set such that each vertex is either in D or has a neighbour in D. In this paper, we are interested in the enumeration of (inclusion-wise) minimal dominating sets in graphs, called the Dom-Enum problem. It is well known that this problem can be polynomially reduced to the Trans-Enum problem in hypergraphs, i.e., the problem of enumerating all minimal transversals in a hypergraph. Firstly we show that the Trans-Enum problem can be polynomially reduced to the Dom-Enum problem. As a consequence there exists an output-polynomial time algorithm for the Trans-Enum problem if and only if there exists one for the Dom-Enum problem. Secondly, we study the Dom-Enum problem in some graph classes. We give an output-polynomial time algorithm for the Dom-Enum problem in split graphs, and introduce the completion of a graph to obtain an output-polynomial time algorithm for the Dom-Enum problem in P_6-free chordal graphs, a proper superclass of split graphs. Finally, we investigate the complexity of the enumeration of (inclusion-wise) minimal connected dominating sets and minimal total dominating sets of graphs. We show that there exists an output-polynomial time algorithm for the Dom-Enum problem (or equivalently Trans-Enum problem) if and only if there exists one for the following enumeration problems: minimal total dominating sets, minimal total dominating sets in split graphs, minimal connected dominating sets in split graphs, minimal dominating sets in co-bipartite graphs.

1. Introduction

The Minimum Dominating Set problem is a classic and well-studied graph optimisation problem. *A dominating set* in a graph G is a subset D of its set of vertices such that each vertex is either in D or has a neighbour in D. Computing a minimum dominating set has numerous applications in many areas, *e.g.*, networks, graph theory (see for instance the book [17]). In this paper we are interested in the enumeration of *minimal (connected, total) dominating sets* in graphs.

Enumeration problems have received much interest over the past decades due to their applications in computer science [1, 9, 15, 16, 25]. For these problems the size of the output may be exponential in the size of the input, which in general is different from optimisation or counting problems where the size of the output is polynomially related to the size of the input. A natural parameter for measuring the time complexity of an enumeration algorithm is the sum of the sizes of the input and output. An algorithm whose running time is bounded by a polynomial depending on the sum of the sizes of the input and output is called an *output-polynomial time* algorithm (also called *total-polynomial time* or *output-sensitive* algorithm).

M.M. Kanté and V. Limouzy are supported by the French Agency for Research under the DORSO project.

A. Mary and L. Nourine are partially supported by the French Agency for Research under DEFIS program DAG, ANR-09-DEFIS, 2009-2012.

A preliminary version of Sections 4 and 5 appeared in the proceedings of FCT 2011.

The enumeration of minimal dominating sets of graphs (Dom-Enum problem for short) is closely related to the well-known Trans-Enum problem in hypergraphs, which consists in enumerating the set of minimal *transversals* (or *hitting sets*) of a hypergraph. A transversal of a hypergraph is a subset of its ground set which has a non empty intersection with every hyperedge. One can notice that the set of minimal dominating sets of a graph is in bijection with the set of minimal transversals of its *closed neighbourhood hypergraph* [7]. The Trans-Enum problem has been intensively studied due to its connections to several problems in such fields as data-mining and learning [11, 12, 16, 20, 24]. It is still open whether there exists an output-polynomial time algorithm for the Trans-Enum problem, but several classes where an output-polynomial time algorithm exists have been identified (see for instance the survey [13]). So, classes of graphs whose closed neighbourhood hypergraphs are in one of these identified classes of hypergraphs admit also output-polynomial time algorithms for the Dom-Enum problem. Examples of such graph classes are planar graphs and bounded degree graphs (see [18, 19] for more information). Recently, the Dom-Enum problem has been studied by several groups of authors [8, 14]. Their research on exact exponential-time algorithms triggered a new approach to the design of enumeration algorithms which uses classical worst-case running time analysis, i.e., the running time depends on the length of the input.

In this paper, we first prove that the Trans-Enum problem can be polynomially reduced to the Dom-Enum problem. Since the other direction also holds, the two problems are *equivalent*, *i.e.*, there exists an output-polynomial time algorithm for the Dom-Enum problem if and only if there exists one for the Trans-Enum problem. One could possibly expect to benefit from graph theory tools to solve the two problems and at the same time many other enumeration problems equivalent to the Trans-Enum problem (see [11] for examples of problems equivalent to Trans-Enum). In addition, we show that there exists an output-polynomial time algorithm for the Dom-Enum problem (or equivalently Trans-Enum problem) if and only if there exists one for the following enumeration problems: TDom-Enum problem, CDom-Enum in split graphs, TDom-Enum in split graphs, Dom-Enum in co-bipartite graphs, where the TDom-Enum problem corresponds to the enumeration of minimal *total dominating* sets.

We then characterise graphs where the addition of edges changes the set of minimal dominating sets. The maximal extension (addition of edges) that keeps invariant the set of minimal dominating sets can be computed in polynomial time, and appears to be a useful tool for getting output-polynomial time algorithms for the Dom-Enum problem in new graph classes such as P_6-free chordal graphs. As a consequence, Dom-Enum in split graphs and Dom-Enum in P_6-free chordal graphs are linear delay and polynomial space.

We finally study the complexity of the enumeration of minimal *connected dominating* sets (called the CDom-Enum problem). The Minimum Connected Dominating Set problem is a well-known and well-studied variant of the Minimum Dominating Set problem due to its applications in networks [17, 28]. We have proved in [18] that CDom-Enum in split graphs is equivalent to the Trans-Enum problem. We will extend this result to other graph classes. Indeed, we prove that the minimal connected dominating sets of a graph are the minimal transversals of its minimal separators. As a consequence, in any class of graphs with a polynomially bounded number of minimal separators, the CDom-Enum problem can be polynomially reduced to the Trans-Enum problem; examples of such classes are chordal graphs, circle graphs and circular-arc graphs [5, 21, 23]. Finally, we show that the CDom-Enum problem is harder than the Dom-Enum problem.

Paper Organisation. Some needed definitions are defined in Section 2. The equivalence between the TRANS-ENUM problem, the DOM-ENUM problem and the TDOM-ENUM problem is given in Section 3. We recall in Section 4 the output-polynomial time algorithm for the DOM-ENUM problem in split graphs published in [18]. Maximal extensions (additions of edges) of graphs are defined in Section 5 and a use of these maximal extensions to obtain an output-polynomial time algorithm for the DOM-ENUM problem in P_6-free chordal graphs is also given. The CDOM-ENUM problem is investigated in Section 6.

2. Preliminaries

If A and B are two sets, $A \backslash B$ denotes the set $\{x \in A \mid x \notin B\}$. The power-set of a set V is denoted by 2^V. We denote by $\mathbb{N}$ the set containing zero and the positive integers. The size of a set A is denoted by $|A|$.

We refer to [10] for graph terminology not defined below; all graphs considered in this paper are undirected, finite and simple. A graph G is a pair $(V(G), E(G))$, where $V(G)$ is the set of vertices and $E(G) \subseteq V(G) \times V(G)$, the set of edges, is symmetric. An edge between x and y is denoted by xy (equivalently yx). The subgraph of G induced by $X \subseteq V(G)$, denoted by $G[X]$, is the graph $(X, E(G) \cap (X \times X))$; $G \backslash X$ is the graph $G[V(G) \backslash X]$. A graph is said to be *chordal* if it has no induced cycle of length greater than or equal to 4; it is a *split* graph if its vertex set can be partitioned into an independent set S and a clique C. Notice that split graphs form a proper subclass of chordal graphs. For two graphs G and H, we say that G is *H-free* if G does not contain H as an induced subgraph. For $k \geqslant 1$, we let P_k be the path on k vertices. For a graph G, we let $N_G(x)$, the set of neighbours of x, be the set $\{y \in V(G) \mid xy \in E(G)\}$, and we let $N_G[x]$ be $N_G(x) \cup \{x\}$. For $X \subseteq V(G)$, we write $N_G[X]$ and $N_G(X)$ for respectively $\bigcup_{x \in X} N_G[x]$ and $N_G[X] \backslash X$.

A *dominating set* in a graph G is a set of vertices D such that every vertex of G is either in D or is adjacent to some vertex of D. It is said to be *minimal* if it does not contain any other dominating set as a subset. The set of all minimal dominating sets of G will be denoted by $\mathcal{D}(G)$. Let D be a dominating set of G and $x \in D$. We say that x has a *private neighbour* y in G if $y \in N_G[x] \backslash N_G[D \backslash \{x\}]$. Note that a private neighbour of a vertex $x \in D$ in G is either x itself, or a vertex in $V(G) \backslash D$, but never a vertex $y \in D \backslash \{x\}$. The set of private neighbours of $x \in D$ in G is denoted by $P_D(x)$. The following is straightforward.

Lemma 1. *Let D be a dominating set of a graph G. Then D is a minimal dominating set if and only if $P_D(x) \neq \emptyset$ for every $x \in D$.*

A *hypergraph* $\mathcal{H}$ is a pair $(V(\mathcal{H}), \mathcal{E}(\mathcal{H}))$ where $V(\mathcal{H})$ is a finite set and $\mathcal{E}(\mathcal{H}) \subseteq 2^{V(\mathcal{H})} \backslash \{\emptyset\}$. It is worth noticing that graphs are special cases of hypergraphs. We will call the elements of $V(\mathcal{H})$ vertices and elements of $\mathcal{E}(\mathcal{H})$ hyperedges, and when the context is clear a hypergraph will be denoted by its set of hyperedges only. If $\mathcal{H}$ is a hypergraph, we let $I(\mathcal{H})$, the *bipartite incidence* graph of $\mathcal{H}$, be the graph with vertex set $V(\mathcal{H}) \cup \{y_e \mid e \in \mathcal{E}(\mathcal{H})\}$ and edge set $\{xy_e \mid x \in V(\mathcal{H}),\ e \in \mathcal{E}(\mathcal{H})$ and $x \in e\}$. Note that the neighbourhood of the vertex y_e in $I(\mathcal{H})$ is exactly the set e. A hypergraph $\mathcal{H}$ is said to be *simple* if

(i) for all $e, e' \in \mathcal{E}(\mathcal{H})$, $e \subseteq e' \Longrightarrow e = e'$, and
(ii) $V(\mathcal{H}) = \bigcup_{e \in \mathcal{E}(\mathcal{H})} e$.

For a hypergraph $\mathcal{H}$ we denote by $Min(\mathcal{H})$ the hypergraph on the same vertex set and keeping only minimal hyperedges, *i.e.*, $\mathcal{E}(Min(\mathcal{H})) := \{e \in \mathcal{E}(\mathcal{H}) \mid \forall e' \in \mathcal{E}(\mathcal{H}) \backslash \{e\},\ e' \nsubseteq e\}$. A *transversal* (or *hitting set*) of $\mathcal{H}$ is a subset of $V(\mathcal{H})$ that has a non-empty intersection with every hyperedge of $\mathcal{E}(\mathcal{H})$; it is *minimal* if it does not contain any other transversal as a subset. The set of all minimal transversals of $\mathcal{H}$ is denoted by $tr(\mathcal{H})$. The size of a hypergraph $\mathcal{H}$, denoted by $\|\mathcal{H}\|$, is $|V(\mathcal{H})| + \sum_{e\in\mathcal{E}(\mathcal{H})} |e|$. The set of all hypergraphs (respectively all graphs) is denoted by $\mathscr{H}$ (respectively $\mathscr{G}$).

Proposition 2 ([3])**.** *For each simple hypergraph $\mathcal{H}$, we have $tr(tr(\mathcal{H})) = \mathcal{H}$.*

From Proposition 2, we obtain the following.

Corollary 3. *For each simple hypergraph $\mathcal{H}$ and each $x \in V(\mathcal{H})$, there exists $T \in tr(\mathcal{H})$ such that $x \in T$.*

An *enumeration algorithm* (algorithm for short) for a set $\mathscr{C}$ is an algorithm that lists the elements of $\mathscr{C}$ without repetitions. Let $\varphi(X)$ be a hypergraph property where X is a subset of vertices (for instance $\varphi(X)$ could be "X is a transversal"). For a hypergraph $\mathcal{H}$, we let $\mathscr{C}_\varphi(\mathcal{H})$ be the set $\{Z \subseteq V(\mathcal{H}) \mid \varphi(Z)$ is true in $\mathcal{H}\}$. An *enumeration problem* for the hypergraph property $\varphi(X)$ takes as input a hypergraph $\mathcal{H}$, and the task is to enumerate, without repetitions, the set $\mathscr{C}_\varphi(\mathcal{H})$. An algorithm for $\mathscr{C}_\varphi(\mathcal{H})$ is an *output-polynomial time* algorithm if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that $\mathscr{C}_\varphi(\mathcal{H})$ is listed in time $p(||\mathcal{H}|| + ||\mathscr{C}_\varphi(\mathcal{H})||)$. Notice that since an algorithm $\mathcal{A}$ for an enumeration problem takes a hypergraph as input and outputs a hypergraph with same vertex set, we can consider it as a function $\mathcal{A} : \mathscr{H} \to \mathscr{H}$. Let $f : \mathbb{N} \to \mathbb{N}$. We say that an algorithm enumerates $\mathscr{C}_\varphi(\mathcal{H})$ with delay $f(||\mathcal{H}||)$ if, after a polynomial time pre-processing, it outputs the elements of $\mathscr{C}_\varphi(\mathcal{H})$ without repetitions, the delay between two outputs being bounded by $f(||\mathcal{H}||)$. If f is a polynomial (or a linear function), we call it a polynomial (or linear) delay algorithm.

Definition 4. *Let P and P' be enumeration problems for hypergraph properties $\varphi(X)$ and $\varphi'(X)$ respectively. We say that P' is* at least as hard *as P, denoted by $P \preceq_{op} P'$, if an output-polynomial time algorithm for P' implies an output-polynomial time algorithm for P.*

Two enumeration problems P and P' are *equivalent* if $P \preceq_{op} P'$ and $P' \preceq_{op} P$. We denote by Trans-Enum the enumeration problem of minimal transversals in hypergraphs. Similarly, we denote by Dom-Enum the enumeration problem of minimal dominating sets in graphs. For a problem P and a subclass $\mathcal{C}$ of instances of P, we denote by $P(\mathcal{C})$ the problem P restricted to the instances in $\mathcal{C}$. For instance, Dom-Enum(split graphs) denotes the problem of enumerating the set of minimal dominating sets in split graphs.

3. Dom-Enum is Equivalent to Trans-Enum

The fact that Dom-Enum $\preceq_{op}$ Trans-Enum can be considered folklore. Let us remind it for completeness. For a graph G, we let $\mathcal{N}(G)$, the *closed neighbourhood hypergraph*, be $(V(G), \{N_G[x] \mid x \in V(G)\})$.

Lemma 5 (Folklore [7])**.** *Let G be a graph and $D \subseteq V(G)$. Then D is a dominating set of G if and only if D is a transversal of $\mathcal{N}(G)$ if and only if D is a transversal of $Min(\mathcal{N}(G))$.*

Corollary 6. Dom-Enum $\preceq_{op}$ Trans-Enum.

Proof. From Lemma 5, we have that $tr(\mathcal{N}(G)) = \mathcal{D}(G)$. Hence, if we have an output-polynomial time algorithm for Trans-Enum then we can use it to enumerate all minimal dominating sets of a graph in output-polynomial time. □

Corollary 7. *Let G be a graph and $x \in V(G)$. Then there exists $D \in \mathcal{D}(G)$ such that $x \in D$.*

Proof. Corollary of Lemma 5 and Corollary 3. □

We now prove that TRANS-ENUM $\preceq_{op}$ DOM-ENUM. One may wonder whether with every hypergraph $\mathcal{H}$ one can associate a graph G such that $\mathcal{D}(G) = tr(\mathcal{H})$. However, the following result shows that such a reduction does not exist.

Proposition 8. *For every function $f : \mathscr{H} \to \mathscr{G}$, there exists $\mathcal{H} \in \mathscr{H}$ such that $tr(\mathcal{H}) \neq \mathcal{D}(f(\mathcal{H}))$.*

Proof. Let $\mathcal{H}$ be a simple hypergraph with $|V(\mathcal{H})| = |\mathcal{E}(\mathcal{H})| = n$ and such that $\mathcal{H}$ is not the closed neighbourhood hypergraph of any graph. Such a hypergraph exists (see for instance [6]). Now assume that there exists a graph G such that $\mathcal{D}(G) = tr(\mathcal{H})$. Note that since each vertex of a simple hypergraph belongs to at least one minimal transversal (Corollary 3), and since each vertex of a graph appears in at least one minimal dominating set (Corollary 7), we have $V(G) = V(\mathcal{H})$. By Lemma 5, $tr(\mathcal{H}) = tr(\mathcal{N}(G)) = tr(Min(\mathcal{N}(G)))$ and so $\mathcal{H} = Min(\mathcal{N}(G))$ (Proposition 2). Furthermore, $Min(\mathcal{N}(G)) \subseteq \mathcal{N}(G)$ and $|Min(\mathcal{N}(G))| = |\mathcal{E}(\mathcal{H})| = n = |\mathcal{N}(G)|$ and so $Min(\mathcal{N}(G)) = \mathcal{N}(G)$. We conclude that $\mathcal{H} = \mathcal{N}(G)$ and then $\mathcal{H}$ is the closed neighbourhood hypergraph of G, which contradicts the assumption. □

Despite the above result, we can polynomially reduce TRANS-ENUM to DOM-ENUM. In order to prove this statement we introduce the *co-bipartite incidence* graph associated with every hypergraph $\mathcal{H}$.

Definition 9. *Let $\mathcal{H}$ be a hypergraph. We associate with $\mathcal{H}$ a co-bipartite incidence graph $B(\mathcal{H})$, defined as follows:*

- $V(B(\mathcal{H})) := V(I(\mathcal{H})) \cup \{v\}$ *with* $v \notin V(I(\mathcal{H}))$,
- $E(B(\mathcal{H})) := E(I(\mathcal{H})) \cup \{vx \mid x \in V(\mathcal{H})\} \cup \{xy \mid x, y \in V(\mathcal{H})\} \cup \{y_e y_{e'} \mid e, e' \in \mathcal{E}(\mathcal{H})\}$.

In other words, $B(\mathcal{H})$ is obtained from $I(\mathcal{H})$ by adding a new vertex that is made adjacent to all vertices in $V(\mathcal{H})$, and replacing the subgraph induced by $V(\mathcal{H})$ (resp. $\{y_e \mid e \in \mathcal{E}(\mathcal{H})\}$) by a clique on the same set; see Figure 1 for an illustration. The following is straightforward to prove.

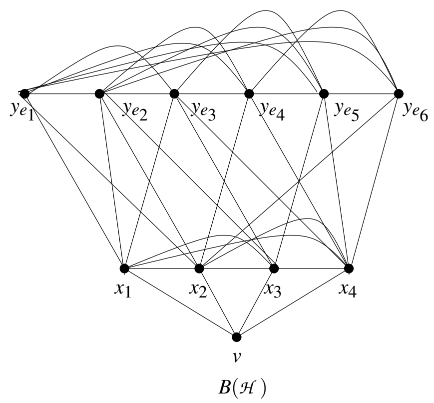

FIGURE 1. An example of the co-bipartite incidence graph $B(\mathcal{H})$ of the hypergraph $\mathcal{H} = (\{x_1, x_2, x_3, x_4\}, \{e_1, e_2, e_3, e_4, e_5, e_6\})$ where $e_1 = \{x_1, x_2\}$, $e_2 = \{x_1, x_2, x_3\}$, $e_3 = \{x_1, x_3, x_4\}$, $e_4 = \{x_2, x_4\}$, $e_5 = \{x_3, x_4\}$, $e_6 = \{x_2, x_4\}$. The set $\{x_1, x_2\}$ is a minimal transversal of $\mathcal{H}$ and a minimal dominating set of $B(\mathcal{H})$.

Lemma 10. *Let $\mathcal{H}$ be a hypergraph and T a transversal of $\mathcal{H}$. Then T is a dominating set of $B(\mathcal{H})$.*

The following lemma claims that there is only a quadratic number of minimal dominating sets of $B(\mathcal{H})$ that are not minimal transversals of $\mathcal{H}$.

Lemma 11. *Let $\mathcal{H}$ be a hypergraph and let D be a minimal dominating set of $B(\mathcal{H})$. Then D is either equal to $\{x, y_e\}$ with $x \in V(\mathcal{H}) \cup \{v\}$ and $e \in \mathcal{E}(\mathcal{H})$ or D is a minimal transversal of $\mathcal{H}$.*

Proof. As v must be dominated by D, $D \cap (V(\mathcal{H}) \cup \{v\}) \neq \emptyset$. Let $x \in D \cap (V(\mathcal{H}) \cup \{v\})$. Assume that $D \cap \{y_e \mid e \in \mathcal{E}(\mathcal{H})\} \neq \emptyset$. Since D is a minimal dominating set, x dominates $V(\mathcal{H}) \cup \{v\}$, and since $\{y_e \mid e \in \mathcal{E}(\mathcal{H})\}$ is a clique, $|D \cap \{y_e \mid e \in \mathcal{E}(\mathcal{H})\}| = 1$. This implies that D is of the form $\{x, y_e\}$. So assume that $D \subseteq V(\mathcal{H}) \cup \{v\}$. It is easy to see that $D \subseteq V(\mathcal{H})$, because if v is in D, since $N_G[v] \cap \{y_e \mid e \in \mathcal{E}(\mathcal{H})\} = \emptyset$ and $\mathcal{H}$ contains at least one non-empty hyperedge, D must contain another vertex x from $V(\mathcal{H})$. But, since $N_G[v] \subseteq N_G[x]$, $P_D(v) = \emptyset$ which contradicts the minimality of D (cf. Lemma 1). We now show that such a D is a transversal of $\mathcal{H}$. Indeed since D is included in $V(\mathcal{H})$, every vertex y_e with $e \in \mathcal{E}(\mathcal{H})$ must be incident with a vertex in $D \cap V(\mathcal{H})$ and then D is a transversal of $\mathcal{H}$. Lemma 10 ensures that D is a minimal transversal. □

Theorem 12. Trans-Enum $\preceq_{op}$ Dom-Enum*(co-bipartite graphs).*

Proof. Assume there exists an output-polynomial time algorithm $\mathcal{A}$ for the Dom-Enum problem which, given a co-bipartite graph G, outputs $\mathcal{D}(G)$ in time $p(||G|| + |\mathcal{D}(G)|)$ where p is a polynomial. Given a hypergraph $\mathcal{H}$, we construct the co-bipartite graph $B(\mathcal{H})$ and call $\mathcal{A}$ on $B(\mathcal{H})$. By Lemma 11, $\mathcal{A}$ on $B(\mathcal{H})$ outputs all minimal transversals of $\mathcal{H}$. We now discuss the time complexity. We clearly have $||B(\mathcal{H})|| = O(||\mathcal{H}||)$ and $B(\mathcal{H})$ can be constructed in time $O(||\mathcal{H}||)$. Moreover by Lemma 11, $||\mathcal{D}(B(\mathcal{H}))|| \leqslant ||tr(\mathcal{H})|| + |V(\mathcal{H})| \times |\mathcal{E}(\mathcal{H})|$. Therefore, $\mathcal{A}$ on $B(\mathcal{H})$ runs in time $O(p(||\mathcal{H}|| + ||tr(\mathcal{H})||) + |V(\mathcal{H})| \times |\mathcal{E}(\mathcal{H})|)$, which is polynomial on $||\mathcal{H}|| + ||tr(\mathcal{H})||$. □

Corollary 6 and Theorem 12 together imply the following result.

Corollary 13. Dom-Enum*(co-bipartite graphs),* Dom-Enum *and* Trans-Enum *are all equivalent.*

From Corollary 13, we can deduce some equivalences between Dom-Enum and some other enumeration problems. For instance, a *total dominating set* is a dominating set D such that the subgraph induced by D contains no isolated vertex. We call TDom-Enum the enumeration problem of (inclusion-wise) minimal total dominating sets. To prove the next lemma we associate with every hypergraph a split-incidence graph.

Definition 14. *The split-incidence graph $I'(\mathcal{H})$ associated with a hypergraph $\mathcal{H}$ is the graph obtained from $I(\mathcal{H})$ by turning the independent set corresponding to $V(\mathcal{H})$ into a clique (see Figure 2). The resulting graph is a split graph.*

Lemma 15. TDom-Enum*(split graphs),* Trans-Enum *and* TDom-Enum *are all equivalent.*

Proof. It is enough to prove that TDom-Enum $\preceq_{op}$ Trans-Enum and Trans-Enum $\preceq_{op}$TDom-Enum(split graphs) since TDom-Enum(split graphs) $\preceq_{op}$ TDom-Enum. We first show that TDom-Enum $\preceq_{op}$ Trans-Enum (the reduction was first noted by [27]). For a graph G, we let $\mathcal{N}_o(G) := (V(G), \{N_G(x) \mid x \in V(G)\})$, the *open neighbourhood hypergraph.* We claim that $\mathcal{TD}(G) = tr(\mathcal{N}_o(G))$ where $\mathcal{TD}(G)$

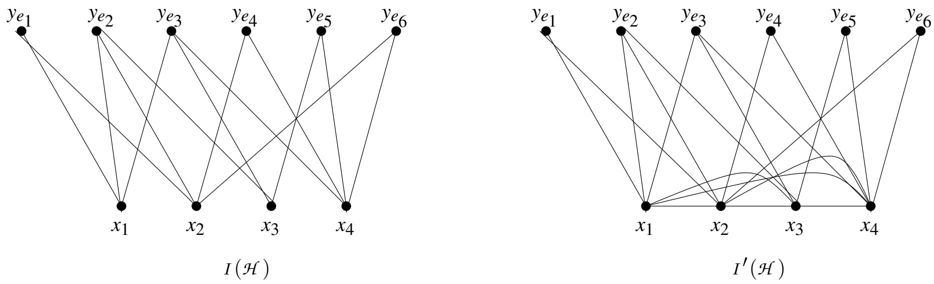

FIGURE 2. An example of the bipartite incidence graph $I(\mathcal{H})$ and the split-incidence graph $I'(\mathcal{H})$ of the hypergraph in Figure 1.

denotes the set of minimal total dominating sets of G. Let G be a graph. It is easy to see that $D \subseteq V(G)$ is a total dominating set in G if and only if it is a transversal of $\mathcal{N}_o(G)$. Indeed, if D is a total dominating set of G, then for each $x \in V(G)$, $N_G(x) \cap D \neq \emptyset$. Therefore, D is a transversal of $\mathcal{N}_o(G)$. Conversely, if T is a transversal of $\mathcal{N}_o(G)$, then for each $x \in V(G)$, $T \cap N_G(x) \neq \emptyset$, *i.e.*, T is a total dominating set of G.

We now show that TRANS-ENUM $\preceq_{op}$TDOM-ENUM(split graphs). Let $\mathcal{H}$ be a hypergraph. Assume furthermore that $\mathcal{H}$ has no dominating vertex, *i.e.*, a vertex belonging to all edges. Note that this case is not restrictive since if $x \in V(\mathcal{H})$ is a dominating vertex, then $tr(\mathcal{H}) = \{x\} \cup tr(\mathcal{H}\backslash\{x\})$ and we can consider this reduced hypergraph. We now show that $\mathcal{TD}(I'(\mathcal{H})) = tr(\mathcal{H})$.

(i) Let D be a minimal total dominating set of $I'(\mathcal{H})$, and let $e \in \mathcal{E}(\mathcal{H})$. Then there exists $x \in V(\mathcal{H}) \cap D$ such that $xy_e \in E(I'(\mathcal{H}))$, *i.e.*, $x \in e$. We now claim that $y_e \notin D$ for all $e \in \mathcal{E}(\mathcal{H})$. Otherwise, there exists $x \in e \cap D$ and since $I'(\mathcal{H})[V(\mathcal{H})]$ is a clique, $D\backslash\{y_e\}$ is also a total dominating set, contradicting the minimality of D. Thus D is a transversal of $\mathcal{H}$.

(ii) Let T be a transversal of $\mathcal{H}$. Then for all $e \in \mathcal{E}(\mathcal{H})$, $T \cap e \neq \emptyset$, *i.e.*, for all $z \in V(I'(\mathcal{H}))\backslash V(\mathcal{H})$, there exists $x \in T$ such that $xz \in E(I'(\mathcal{H}))$. Since there is no dominating vertex, $|T| \geqslant 2$, and because $I'(\mathcal{H})[V(\mathcal{H})]$ is a clique, for all $x \in V(\mathcal{H})$, there exists $y \in T$ such that $xy \in E(I'(\mathcal{H}))$. Hence, T is a total dominating set of $I'(\mathcal{H})$.

From (i) and (ii) we can conclude that $\mathcal{TD}(I'(\mathcal{H})) = tr(\mathcal{H})$. □

As a corollary of Lemma 15 and Corollary 13 we get the following.

Corollary 16. DOM-ENUM *and* TDOM-ENUM *are equivalent.*

These results may enable new approaches to consider the TRANS-ENUM problem as a graph problem. We will give some evidence in the following sections. We conclude this section by stating the following decision problem DOM-GRAPH that arises from Corollary 13 and seems to be interesting on its own.

Input. A hypergraph $\mathcal{H}$ and a positive integer k.
Output. Does there exist a graph G and a set $F \subseteq 2^{V(G)}$ with $|F| \leqslant k$ and such that $\mathcal{D}(G) = tr(\mathcal{H}) \cup F$?

It is an NP-complete problem because the problem of *realisability* of a hypergraph is a special case with $k = 0$ [6]. For $k = |V(\mathcal{H})| \cdot |\mathcal{E}(\mathcal{H})|$, the DOM-GRAPH problem can be solved in polynomial time by Corollary 13. We leave open its complexity for $1 \leqslant k < |V(\mathcal{H})| \cdot |\mathcal{E}(\mathcal{H})|$.

4. Dom-Enum in Split Graphs

We recall that a graph G is a *split* graph if its vertex set can be partitioned into an independent set S and a clique C. Here we consider S to be maximal. We will denote a split graph G by the pair $(C(G) \cup S(G), E(G))$. We prove in this section that Dom-Enum(split graphs) admits a linear delay algorithm that uses polynomial space. A minimal dominating set D of a split graph G can be partitioned into a clique and an independent set, denoted respectively by $D_C := D \cap C(G)$ and $D_S := D \cap S(G)$. Lemma 17 shows that a minimal dominating set D of a split graph is characterised by D_C. Note that D_S cannot characterise D, since several minimal dominating sets can have the same set D_S.

Lemma 17. *Let G be a split graph and D a minimal dominating set of G. Then $D_S = S(G)\backslash N_G(D_C)$.*

Lemma 18. *Let $A \subseteq C(G)$. If every element in A has a private neighbour then $A \cup (S(G)\backslash N_G(A))$ is a minimal dominating set of G.*

Proof. It is clear that $A \cup (S(G)\backslash N_G(A))$ is a dominating set. To see why it is minimal, it suffices to observe that every vertex $s \in S(G)\backslash N_G(A)$ has at least one private neighbour, namely vertex s itself. Note that every vertex in A has a private neighbour by assumption. Hence, $A \cup (S(G)\backslash N_G(A))$ is a minimal dominating set due to Lemma 1. □

Lemma 19. *Let D be a minimal dominating set of a split graph G. Then for all $A \subseteq D_C$, the set $A \cup (S(G)\backslash N_G(A))$ is a minimal dominating set of G.*

Proof. Let D be a minimal dominating set of G and let $A \subseteq D_C$. Clearly each $x \in A$ has a private neighbour since D is a minimal dominating set. According to Lemma 18, $A \cup (S(G)\backslash N_G(A))$ is a minimal dominating set of G. □

A consequence of Lemmas 17 and 18 is the following.

Corollary 20. *Let G be a split graph. Then there is a bijection between $\mathcal{D}(G)$ and the set $\{A \subseteq C(G) \mid \forall x \in A, x$ has a private neighbour$\}$.*

We now describe an algorithm, which we call *DominantSplit*, that takes as input a split graph G with a linear ordering $\sigma : V(G) \to \{1, \ldots, |V(G)|\}$ of its vertex set and a minimal dominating set D of G, and outputs all minimal dominating sets Q of G such that $D_C \subseteq Q_C$. Then, whenever $D = S(G)$, the algorithm enumerates all minimal dominating sets of G. The algorithm starts by computing the largest vertex y (with respect to the linear ordering σ) in D_C. Then, the algorithm checks whether the set D_C can be extended, i.e. whether there exists a vertex $x \in C(G)\backslash D_C$ which is greater than y and such that every vertex in $D_C \cup \{x\}$ has a private neighbour. For each such x, the algorithm builds the minimal dominating set D' such that $D'_C = D_C \cup \{x\}$ (which is unique by Lemma 17) and recursively calls the algorithm on D'. The pseudo-code is given in Algorithm 1.

Theorem 21. *Let G be a split graph with n vertices and m edges and let σ be any linear ordering of $V(G)$. Then DominantSplit$(G, \sigma, S(G))$ enumerates the set $\mathcal{D}(G)$ with $O(n+m)$ delay and uses space bounded by $O(n^2)$.*

Proof. We first prove the correctness of the algorithm. We first prove that each minimal dominating set is listed once.

We prove the completeness using induction on the number of elements in the clique $C(G)$. First the only minimal dominating set D of G such that $|D_C| = 0$ is $S(G)$ which corresponds to the first call of the algorithm. Indeed, if $D \cap C = \emptyset$ then each vertex of $S(G)$ must belong to D to dominate itself. Moreover, by Lemma

Algorithm 1: *DominantSplit*(G, σ, D)

Input: A split graph $G = (C(G) \cup S(G), E(G))$, a linear ordering $\sigma : V(G) \to \{1, \ldots, |V(G)|\}$ and a minimal dominating set D of G.

begin

  output (D)

  Cov $= \varnothing$

1 Let $y \in D_C$ be such that $\sigma(y) = max\{\sigma(x) \mid x \in D_C\}$

2 **foreach** $x \in C(G) \backslash D_C$ *and* $\sigma(x) > \sigma(y)$ **do**

3   **if** *each vertex in* $D_C \cup \{x\}$ *has a private neighbour* **then**

4   **Cov** = **Cov** $\cup \{x\}$

5 **foreach** $x \in$ **Cov do**

6   *DominantSplit*$(G, \sigma, D_C \cup \{x\} \cup (S(G) \backslash N_G(D_C \cup \{x\})))$

end

18 it is a minimal dominating set. Assume now that every $D' \in \mathcal{D}(G)$ such that $|D'_C| \leqslant k$, is returned by the algorithm and let D be a minimal dominating set such that $|D_C| = k + 1$. Let x be the greatest vertex of D_C (with respect to σ). By Lemma 19, $D' := D_C \backslash \{x\} \cup (S(G) \backslash N_G(D_C \backslash \{x\}))$ is a minimal dominating set of G. Furthermore, $|D'_C| = |D_C \backslash \{x\}| = k$, and then D' is returned by the algorithm (by the inductive hypothesis). Note also that since $D \in \mathcal{D}(G)$, every vertex in $D'_C \cup \{x\} = D_C$ has a private neighbour, and since x is greater than all vertices of D'_C (w.r.t. σ), x is added to **Cov** by the algorithm. Then in the next step *DominantSplit*$(G, \sigma, D'_C \cup \{x\} \cup (S(G) \backslash N_G(D'_C \cup \{x\}))))$ will be called and then $D'_C \cup \{x\} \cup (S(G) \backslash N_G(D'_C \cup \{x\})) = D_C \cup (S(G) \backslash N_G(D_C))$ will be returned, which is equal to D by Lemma 17.

Now let us show that if a set A is returned, then A is a minimal dominating set of G. We have two cases: either $A = S(G)$ (which corresponds to the first call) or $A = D_C \cup \{x\} \cup (S(G) \backslash N_G(D_C \cup \{x\}))$. Clearly if $A = S(G)$ then A is a minimal dominating set. Now if $A = D_C \cup \{x\} \cup (S(G) \backslash N_G(D_C \cup \{x\}))$ then every element in $D_C \cup \{x\}$ has a private neighbour (cf. Line 3 of algorithm 1). Using Lemma 18 we conclude that A is a minimal dominating set.

Moreover each minimal dominating set is listed exactly once. Indeed, a minimal dominating set D' is obtained by a call *DominantSplit*$(G, \sigma, D_C \cup \{x\} \cup (S(G) \backslash N_G(D_C \cup \{x\})))$ where D is the minimal dominating set such that $D_C = D'_C \backslash \{x\}$ with x the greatest vertex in D'_C (with respect to σ), which is unique by Lemma 17.

We now discuss the delay and space. The delay between the output of D and the next output is dominated by the time needed to check if any element in $D_C \cup \{x\}$ has a private neighbour.

To do so, we use an array marks[1..n] initialised to 0, and for each element in $D_C \cup \{x\}$ we increase the marks of its neighbours by 1. To check that every element y in $D_C \cup \{x\}$ has a private neighbour, it suffices to check that y has at least a neighbour with mark 1. Note that we check only neighbourhood in the stable $S(G)$. This can be done in time $O(n+m)$. Since the depth of the recursive tree is at most n and at each node we store the set **Cov**, the space memory is bounded by $O(n^2)$. $\square$

5. Completion

In this section we introduce the notion of the maximal extension of a graph by keeping the set of minimal dominating sets invariant. The idea behind this operation is to maintain invariant the minimal hyperedges, with respect to inclusion, in $\mathcal{N}(G)$.

For a graph G we denote by $IR(G)$ the set of vertices (called *irredundant* vertices) that are minimal with respect to the neighbourhood inclusion. In case of equality between minimal vertices, exactly one is considered as irredundant. All the other vertices are called *redundant* and the set of redundant vertices is denoted by $RN(G)$. The *completion* graph of a graph G is the graph G_{co} with vertex set $V(G)$ and edge set $E(G) \cup \{xy \mid x, y \in RN(G),\ x \neq y\}$, *i.e.*, G_{co} is obtained from G by adding precisely those edges to G that make $RN(G)$ into a clique. Note that the completion graph of a split graph G is G itself, since all vertices in $S(G)$ are irredundant. However, the completion operation does not preserve the chordality of a graph. For instance, trees are chordal graphs but their completion graphs are not always chordal. Figure 3 gives some examples of completion graphs.

Remark 22. *Note that if a vertex x is redundant, then there exists an irredundant vertex y such that $N_G[y] \subseteq N_G[x]$. Indeed since x is redundant, the set $F := \{z \in V \mid N_G[z] \subseteq N_G[x]\}$ is not empty. Hence, any minimal (with respect to neighbourhood inclusion) vertex y from F is an irredundant vertex.*

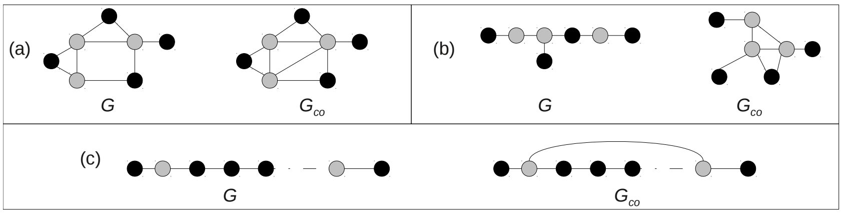

Figure 3. (a) a non-chordal graph whose completion is a split graph (b) a chordal graph with an induced P_6 whose completion is a split graph (c) a path P_n whose completion is not chordal. Redundant vertices are represented in grey.

Proposition 23. *For any graph G, we have $\mathcal{D}(G) = \mathcal{D}(G_{co})$.*

Proof. Let D be a dominating set of a graph G. Since $E(G) \subseteq E(G_{co})$, D is also a dominating set of G_{co}. Now suppose that D is a dominating set of G_{co} and let $x \in V(G)$. If $x \in IR(G)$, then $N_G[x] = N_{G_{co}}[x]$, hence $D \cap N_G[x] \neq \emptyset$. If $x \in RN(G)$, then, due to Remark 22, there exists $y \in IR(G)$ such that $N_G[y] \subseteq N_G[x]$. Hence $D \cap N_G[y] \subseteq D \cap N_G[x] \neq \emptyset$. Therefore, D is a dominating set of G. Since G and G_{co} have the same dominating sets, we deduce that $\mathcal{D}(G) = \mathcal{D}(G_{co})$. ☐

The following proposition claims the optimality of the completion in the sense that no other edges can be added to the graph without changing the set of minimal dominating sets.

Proposition 24. *Let G be a graph and let G' be $(V(G), E(G) \cup \{e\})$ with e a non-edge of G. Then $\mathcal{D}(G) \neq \mathcal{D}(G')$ if and only if $e \cap IR(G) \neq \emptyset$,*

Proof. Consider $\mathcal{N}'(G) := \{N_G[v] \mid v \in IR(G)\}$ and $\mathcal{N}'(G') := \{N_{G'}[v] \mid v \in IR(G')\}$. By the definition of irredundant vertices, for every $u, v \in IR(G)$, we have

$N_G[u] \subseteq N_G[v]$ implies that $u = v$ and therefore $N_G[u] = N_G[v]$. Hence $\mathcal{N}'(G)$ and $\mathcal{N}'(G')$ are simple and correspond respectively to $Min(\mathcal{N}(G))$ and $Min(\mathcal{N}(G'))$.

Let $e := xy$ such that $e \cap IR(G) \neq \emptyset$, and assume without loss of generality that $x \in IR(G)$. Assume that x is still irredundant in G', *i.e.*, $x \in IR(G')$. Then since $y \in N_{G'}[x]$ and $y \notin N_G[x]$, $\mathcal{N}'(G') \neq \mathcal{N}'(G)$. Moreover, thanks to Lemma 5, we have $\mathcal{D}(G) = tr(\mathcal{N}'(G))$ and $\mathcal{D}(G') = tr(\mathcal{N}'(G'))$, and since $\mathcal{N}'(G)$ and $\mathcal{N}'(G')$ are simple, we have $\mathcal{D}(G') \neq \mathcal{D}(G)$ (see Proposition 2). Assume now that $x \in RN(G')$. Hence, $N_G[x] \notin \mathcal{N}'(G')$ and since $N_G[x] \in \mathcal{N}'(G)$, we have $\mathcal{N}'(G) \neq \mathcal{N}'(G')$.

Assume now that $e \cap IR(G) = \emptyset$, i.e. $e \subseteq RN(G)$. Then $IR(G) = IR(G')$ and for all $v \in IR(G)$, $N_G[v] = N_{G'}[v]$. Thus we have $Min(\mathcal{N}(G)) = \mathcal{N}'(G) = \mathcal{N}'(G') = Min(\mathcal{N}(G'))$ and then $\mathcal{D}(G) = tr(\mathcal{N}(G)) = tr(\mathcal{N}(G')) = \mathcal{D}(G')$. □

We now show how to use completion to get an output-polynomial time algorithm for the DOM-ENUM problem restricted to P_6-free chordal graphs. Let us notice that this class properly contains the class of split graphs. The results that follow were already published in [18] without proofs. A vertex is *simplicial* if the graph induced by its neighbourhood is a clique.

Proposition 25. *If G is a P_6-free chordal graph, then for all $x \in IR(G)$, x is a simplicial vertex in G_{co}. Furthermore, the set $IR(G)$ is an independent set in G_{co}.*

Proof. We first show that for all $x \in IR(G)$, x is a simplicial vertex in G_{co}. Assume that there exists $x \in IR(G)$ such that x is not a simplicial vertex in G_{co}. Then there exist $y, z \in N_{G_{co}}[x]$ such that $yz \notin E(G_{co})$. Since x is irredundant in G, there exist $y' \in N_G[y] \backslash N_G[x]$ and $z' \in N_G[z] \backslash N_G[x]$. Observe that $y' \neq z'$, otherwise $\{x, y, y', z\}$ forms an induced C_4 in G. Moreover, since $yz \notin E(G_{co})$, either $z \notin RN(G)$ or $y \notin RN(G)$. Assume without loss of generality that $y \notin RN(G)$. Then $N_G[y'] \nsubseteq N_G[y]$ and so there exists $y'' \in N_G[y'] \backslash N_G[y]$. But then $P := z'zxyy'y''$ forms an induced P_6, because all possible edges between two non consecutive vertices of P would create an induced cycle of length greater than four, contradicting the chordality of G.

We finally show that $IR(G)$ is an independent set in G_{co}. Suppose that there exists $xy \in E(G_{co})$ with $x, y \in IR(G)$. Since for all $z \in IR(G)$, z is a simplicial vertex in G_{co}, it follows that both $N_{G_{co}}[x]$ and $N_{G_{co}}[y]$ are cliques. But since $xy \in E(G_{co})$, we have $N_{G_{co}}[x] = N_{G_{co}}[y]$, otherwise there must exist $z \in N_{G_{co}}[x] \backslash y$ and $yz \notin E(G_{co})$ which is impossible since x is simplicial (by the first statement). Since no edges are added incident with x or y when G_{co} is obtained from G, we must have $N_G[x] = N_G[y]$ contradicting the assumption that x and y are irredundant. □

A consequence of Proposition 25 is the following.

Proposition 26. *Let G be a P_6-free chordal graph. Then G_{co} is a split graph.*

Proof. From Proposition 25, it follows that $IR(G)$ forms an independent set in G_{co}, and since $RN(G)$ forms a clique in G_{co}, we are done. □

The next theorem characterises completion graphs that are split.

Proposition 27. *Let G be a graph. Then G_{co} is a chordal graph if and only if G_{co} is a split graph.*

Proof. Since split graphs are chordal graphs, it is enough to prove that if G_{co} is chordal, then it is a split graph. Assume there exists a graph G such that G_{co} is chordal and not a split graph. Since $RN(G)$ forms a clique in G_{co}, there must exist $x_1, x_2 \in IR(G)$ such that $x_1x_2 \in E(G_{co})$. We prove the following claim, which contradicts the fact that G is finite and therefore suffices to prove Proposition 27.

Claim 28. *There exists an infinite sequence $(x_i)_{i\in\mathbb{N}}$ of distinct vertices in $IR(G)$ such that, for all i, x_i is connected to x_{i+1} and x_{i-1}, and for $j \notin \{i-1, i+1\}$, $x_ix_j \notin E(G_{co})$.*

Proof of Claim 28. Since $x_1 \in IR(G)$, there exists $x'_2 \in N_G[x_2]\backslash N_G[x_1]$. In the same way, there exists $x'_1 \in N_G[x_1]\backslash N_G[x_2]$.
Case 1. $x'_1 \in RN(G)$ and $x'_2 \in RN(G)$. Then $x'_1x'_2 \in E(G_{co})$ and so $C := x_1x_2x'_2x'_1$ forms an induced C_4 of G_{co}, which contradicts the assumptions.
Case 2. $x'_1 \in RN(G)$ and $x'_2 \in IR(G)$. Let $x_3 = x'_2$. We prove by induction that for all $j \geqslant 3$, there exists an induced path $x_1 \dots x_j$ of elements of $IR(G)$. For $j = 3$ the property holds since $x_1x_2x_3$ forms an induced path.

Assume that the property holds for all $j \leqslant k$, in other words, we have a sequence $P := x_1x_2 \dots x_k$ of k distinct elements of $IR(G)$ forming an induced path in G_{co}. We show now that there exists $x_{k+1} \in IR(G)$ such that $x_{k+1}x_k \in E(G_{co})$ and for all $j \leqslant k$, $x_{k+1}x_j \notin E(G_{co})$. Since $x_{k-1} \in IR(G)$, there exists a vertex in $N_G[x_k]\backslash N_G[x_{k-1}]$. We choose x_{k+1} to be such a vertex. Note that $x_{k+1} \notin \{x_k, x_{k-1}\}$ since $x_{k+1} \in N_G[x_k]\backslash N_G[x_{k-1}]$, by definition, and $x_{k+1} \notin \{x_1, ..., x_{k-1}\}$ since $x_1 \dots x_k$ is an induced path and x_{k+1} is adjacent to x_k. In other words x_{k+1} is distinct from x_j for all $j \leqslant k$. Also note that x_{k+1} cannot belong to $RN(G)$, as otherwise $x_{k+1}x'_1x_1 \dots x_k$ would be a cycle of length greater than four, contradicting the assumption that G_{co} is chordal. Since $x_{k+1} \in N_{G_{co}}[x_k]\backslash N_{G_{co}}[x_{k-1}]$, if there exists $j < k-1$ with $x_jx_{k+1} \in E(G_{co})$ then $x_j \dots x_{k+1}$ induces a cycle of length at least four. This contradiction finished the proof of Case 2.

Case 3. $x'_1 \in IR(G)$ and $x'_2 \in IR(G)$. Case 3 is identical to Case 2 up to symmetry. □

□

We can now state the following theorem which generalises Theorem 21 to P_6-free chordal graphs. Actually, P_6-free chordal graphs properly contain split graphs, since split graphs are P_5-free chordal graphs.

Theorem 29. *There exists an $O(n+m)$ delay algorithm for the* Dom-Enum *problem in P_6-free chordal graphs with space complexity $O(n^2)$.*

Proof. Let G be a P_6-free chordal graph. First, construct the graph G_{co}, which can clearly be done in polynomial time. Then, enumerate all minimal dominating sets of G_{co}, which can be done with linear delay in the size of G (since the added edges in the completion are not considered by Algorithm 1) and using $O(n^2)$ space due to Theorem 21. The observation that this set coincides with the set of all minimal dominating sets of G due to Proposition 23 finishes the proof of Theorem 29. □

6. Connected Dominating Sets

We investigate in this section the complexity of the enumeration of minimal *connected dominating sets* of a graph. A *connected dominating set* is a dominating set D such that the subgraph induced by D is connected; it is minimal if for each $x \in D$, either $D\backslash\{x\}$ is not a dominating set or the subgraph induced by $D\backslash\{x\}$ is not connected. We denote by CDom-Enum the enumeration problem of minimal connected dominating sets, and by $\mathcal{CD}(G)$ the set of minimal connected dominating sets of a graph G.

Proposition 30 ([18])**.** *For every hypergraph $\mathcal{H}$, $tr(\mathcal{H}) = \mathcal{CD}(I'(\mathcal{H}))$. Hence,* CDom-Enum*(split graphs) is equivalent to* Trans-Enum.

Proof. (i) Let $D \in \mathcal{CD}(I'(\mathcal{H}))$ (cf. Definition 14). Note that every minimal connected dominating set in a split graph is a subset of the clique (cf. [2]) and thus

$D \subseteq V(\mathcal{H})$. Now, for each $e \in \mathcal{E}(\mathcal{H})$, there exists $x \in D$ such that $xy_e \in E(I'(\mathcal{H}))$, hence $D \cap e \neq \emptyset$. And so D is a transversal of $\mathcal{H}$.

(ii) Let T be a transversal of $\mathcal{H}$. Since $I'(\mathcal{H})[V(\mathcal{H})]$ is a clique, T is connected, and for each $x \in V(\mathcal{H})$, there exists $y \in T$ such that $xy \in E(I'(\mathcal{H}))$. Furthermore, for each $e \in \mathcal{E}(\mathcal{H})$, $T \cap e \neq \emptyset$, *i.e.*, for each $y_e \in V(I'(\mathcal{H}))\backslash V(\mathcal{H})$, there is $z \in T$ such that $zy_e \in E(I'(\mathcal{H}))$. Hence, T is a connected dominating set of $I'(\mathcal{H})$.

From (i) and (ii) we can conclude that $\mathcal{CD}(I'(\mathcal{H})) = tr(\mathcal{H})$.

It remains to reduce CDOM-ENUM to TRANS-ENUM. For a split graph G, we let $\mathcal{H}$ be the hypergraph $(C(G), \{N_G(x) \mid x \in S(G)\})$. It is easy to see that $G = I'(\mathcal{H})$ and so from above, $\mathcal{CD}(I'(\mathcal{H})) = tr(\mathcal{H})$. □

We will extend this result to other graph classes and we expect that it is a first step for classifying the complexity of the CDOM-ENUM problem.

A subset $S \subseteq V(G)$ of a connected graph G is called a *separator* of G if $G\backslash S$ is not connected; S is minimal if it does not contain any other separator. Note that this notion is different from the classical notion of minimal *ab-separators*. For two vertices a and b, an *ab-separator* is a subset $S \subseteq V(G)\backslash\{a, b\}$ which disconnects a from b; it is said to be minimal if no proper subset of S disconnects a from b. Every minimal separator is an ab-separator for some pair of vertices a, b. The minimal separators are exactly the minimal ab-separators which do not contain any other cd-separator. For this reason they are often called the *inclusion minimal separators*. Notice that a graph may have an exponential number of minimal separators, but one can enumerate them in output-polynomial time [26]. Algorithms that enumerate all the minimal ab-separators of a graph can be found in [4, 22, 26]. We define $\mathcal{S}(G)$ as the hypergraph $(V(G), \{S \subset V(G) \mid S \text{ is a minimal separator of } G\})$.

Proposition 31. *For every graph G, $\mathcal{CD}(G) = tr(\mathcal{S}(G))$.*

Proof. We first prove that a connected dominating set of G is a transversal of $\mathcal{S}(G)$. Let D be a connected dominating set and assume that there exists a separator S for which $S \cap D = \emptyset$. Let $G_1, \ldots, G_p$ be the connected components of $G[V\backslash S]$. Since D is connected, it must be included in $V(G_i)$ for some $1 \leqslant i \leqslant p$. Assume without loss of generality that $D \subseteq V(G_1)$ and let $x \in V(G_2)$. Then we have $N_G[x] \subseteq V(G_2) \cup S$ and then $N_G[x] \cap D = \emptyset$ which contradicts the fact that D is a dominating set of G.

We now prove that a transversal of $\mathcal{S}(G)$ is a connected dominating set of G. Let T be a transversal of $\mathcal{S}(G)$. We first show that T is a dominating set of G. Suppose not and let N be the set of vertices not covered by T, *i.e.*, $N := \{x \in V(G) \mid N_G[x] \cap T = \emptyset\}$. Then $V(G) = T \cup N_G(T) \cup N$ and by definition of N, there are no edges between N and T. So $G\backslash N_G(T)$ is not connected, in other words, $N_G(T)$ is a separator of G. Hence, $N_G(T)$ contains a minimal separator S which does not intersect T. This contradicts the fact that T is a transversal of $\mathcal{S}(G)$. It remains to prove that $G[T]$ is connected. Assume, for contradiction, that $G[T]$ is not connected. Then $V(G)\backslash T$ is a separator. But then $V(G)\backslash T$ contains a minimal separator S such that $S \cap T \neq \emptyset$. This contradicts again the fact that T is a transversal of $\mathcal{S}(G)$.

Finally since a set S is a transversal of $\mathcal{S}(G)$ if and only if S is a connected dominating set of G, we have that $tr(\mathcal{S}(G)) = \mathcal{CD}(G)$. □

The following corollary shows that any simple hypergraph is the set of minimal separators for some graph, whereas there exist simple hypergraphs which are not neighbourhood hypergraphs (see [6]).

Corollary 32. *For each simple hypergraph $\mathcal{H}$, there exists a split graph G such that $\mathcal{H} = \mathcal{S}(G)$.*

Proof. By Proposition 31, we have $\mathcal{CD}(I'(\mathcal{H})) = tr(\mathcal{S}(I'(\mathcal{H})))$. So, by Proposition 30, we have $tr(\mathcal{H}) = tr(\mathcal{S}(I'(\mathcal{H})))$, and then by Proposition 2, $\mathcal{H} = \mathcal{S}(I'(\mathcal{H}))$. □

Another consequence of Proposition 31 is the following.

Corollary 33. *If a class of graphs $\mathcal{C}$ has a polynomially bounded number of minimal separators, then* CDom-Enum*($\mathcal{C}$)* $\leq_{op}$ Trans-Enum*. Moreover, if the class $\mathcal{C}$ contains split graphs, then* Trans-Enum *is equivalent to* CDom-Enum*($\mathcal{C}$).*

Proof. Assume that one can solve Trans-Enum in output-polynomial time. Let $G \in \mathcal{C}$. Since the set of all minimal separators of a graph can be enumerated in output-polynomial time and since there is a polynomial number of separators, $\mathcal{S}(G)$ can be computed in time polynomial in $||G||$. Furthermore, the fact that $\mathcal{CD}(G) = tr(\mathcal{S}(G))$ by Proposition 31 achieves the proof of the first statement. The second statement follows from the first statement and Proposition 30. □

Among examples of such graph classes we can cite, without being exhaustive, chordal graphs, trapezoid graphs [5], chordal bipartite graphs [21], and circle and circular arc graphs [23].

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD Conference*, pages 207–216, 1993.

[2] Luitpold Babel and Stephan Olariu. Domination and steiner tree problems on graphs with few p4s. In *WG*, pages 337–350, 1998.

[3] Claude Berge. *Hypergraphs: Combinatorics of Finite Sets*. North-Holland, 1989.

[4] Anne Berry, Jean Paul Bordat, and Olivier Cogis. Generating all the minimal separators of a graph. *Int. J. Found. Comput. Sci.*, 11(3):397–403, 2000.

[5] Hans L. Bodlaender, Ton Kloks, Dieter Kratsch, and Haiko Müller. Treewidth and minimum fill-in on d-trapezoid graphs. *J. Graph Algorithms Appl.*, 2(2), 1998.

[6] Endre Boros, Vladimir Gurvich, and Igor E. Zverovich. Neighborhood hypergraphs of bipartite graphs. *Journal of Graph Theory*, 58(1):69–95, 2008.

[7] Andreas Brandstädt, Van Bang Le, and Jeremy Spinrad. *Graph Classes: A Survey*, volume 3 of *SIAM Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics, 1999.

[8] Jean-François Couturier, Pinar Heggernes, Pim van 't Hof, and Dieter Kratsch. Minimal dominating sets in graph classes: Combinatorial bounds and enumeration. *Theor. Comput. Sci.*, 487:82–94, 2013.

[9] Claire Diehl, Claude Jard, and Jean-Xavier Rampon. Reachability analysis on distributed executions. In *TAPSOFT*, pages 629–643, 1993.

[10] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2005.

[11] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Comput.*, 24(6):1278–1304, 1995.

[12] Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Comput.*, 32(2):514–537, 2003.

[13] Thomas Eiter, Kazuhisa Makino, and Georg Gottlob. Computational aspects of monotone dualization: A brief survey. *Discrete Applied Mathematics*, 156(11):2035–2049, 2008.

[14] Fedor V. Fomin, Fabrizio Grandoni, Artem V. Pyatkin, and Alexey A. Stepanov. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Trans Algorithms*, 5(1):1–17, 2008.

[15] Bernhard Ganter and Rudolf Wille. *Formal concept analysis - mathematical foundations*. Springer, 1999.

[16] Dimitrios Gunopulos, Roni Khardon, Heikki Mannila, and Hannu Toivonen. Data mining, hypergraph transversals, and machine learning. In *PODS*, pages 209–216, 1997.

[17] Teresa W. Haynes, Stephen T. Hedetniemi, and Peter J. Slater. *Fundamentals of Domination in Graphs*, volume 208 of *Pure and Applied Mathematics*. Marcel Dekker, 1998.

[18] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. Enumeration of minimal dominating sets and variants. In Olaf Owe, Martin Steffen, and Jan Arne Telle, editors, *FCT*, volume 6914 of *Lecture Notes in Computer Science*, pages 298–309. Springer, 2011.

[19] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. On the neighbourhood helly of some graph classes and applications to the enumeration of minimal dominating sets. In *ISAAC*, pages 289–298, 2012.

[20] Leonid Khachiyan, Endre Boros, Khaled M. Elbassioni, and Vladimir Gurvich. An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals and its application in joint generation. *Discrete Applied Mathematics*, 154(16):2350–2372, 2006.

[21] Ton Kloks and Dieter Kratsch. Treewidth of chordal bipartite graphs. *J. Algorithms*, 19(2):266–281, 1995.

[22] Ton Kloks and Dieter Kratsch. Listing all minimal separators of a graph. *SIAM J. Comput.*, 27(3):605–613, 1998.

[23] Ton Kloks, Dieter Kratsch, and C. K. Wong. Minimum fill-in on circle and circular-arc graphs. *J. Algorithms*, 28(2):272–289, 1998.

[24] Lhouari Nourine and Jean-Marc Petit. Extending set-based dualization: Application to pattern mining. In *ECAI*, pages IOS Press ed, Montpellier, France, 2012.

[25] Frank Ruskey. *Combinatorial generation*. University of Victoria, Canada, 2003.

[26] Hong Shen and Weifa Liang. Efficient enumeration of all minimal separators in a graph. *Theor. Comput. Sci.*, 180(1-2):169–180, 1997.

[27] Stéphan Thomassé and Anders Yeo. Total domination of graphs and small transversals of hypergraphs. *Combinatorica*, 27(4):473–487, 2007.

[28] Jie Wu and Hailan Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 18(1-3):13–36, 2001.

Clermont-Université, Université Blaise Pascal, LIMOS, CNRS, France
E-mail address: {mamadou.kante,limouzy,mary,nourine}@isima.fr