Enumeration of Minimal Dominating Sets and Variants

Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine

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

Abstract. In this paper, we are interested in the enumeration of minimal dominating sets in graphs. A polynomial delay algorithm with polynomial space in split graphs is presented. We then introduce a notion of maximal extension (a set of edges added to the graph) that keeps invariant the set of minimal dominating sets, and show that graphs with extensions as split graphs are exactly the ones having chordal graphs as extensions. We finish by relating the enumeration of some variants of dominating sets to the enumeration of minimal transversals in hypergraphs.

1 Introduction

In many areas such as data mining, data bases, biology, social networks, etc., people are interested in enumerating a list of objects satisfying some properties [2,23]. For instance, in social networks, for marketing purposes, it can be useful to be able to enumerate the maximal communities, which corresponds in graph theory in enumerating maximal cliques. Classically, an algorithm which scans all possible solutions and outputs the desired solutions can be used. However, such a scenario cannot be used since in many cases, the size of outputs can be much smaller than the number of possible solutions. On the other side, since the size of the output can be huge compared to the size of the input, to measure the efficiency of an enumeration algorithm, the size of the input is not relevant for time complexity, contrary to classical decision problems. A natural parameter for measuring the time complexity of an enumeration algorithm is the number of outputs. Therefore, we will say that an enumeration algorithm runs in *output-polynomial* time if its running time is bounded by a polynomial depending on the number of outputs and the size of the input.

Minimum dominating set problem is a classic graph optimization NP-complete problem. However, contrary to other classic NP-complete graph optimization problems where there exist output-polynomial time algorithm for enumerating maximal (or minimal) solutions, *e.g.* maximal cliques or maximal independent sets [18,19], there is no known output-polynomial time algorithm that enumerates the set of minimal dominating sets of a graph. This paper is motivated by the quest for an output-polynomial time algorithm for the enumeration of minimal dominating sets of graphs (Dom for short). The dominating set problem is related to the well-known transversal problem in hypergraphs. Indeed, the set of minimal dominating sets of a graph is in bijection with the set of minimal transversals of its closed neighbourhood hypergraph [5]. It has been shown in [16] that the enumeration of minimal transversals in hypergraphs (Trans-Hyp for short) can be polynomially reduced to Dom. Trans-Hyp has been intensively studied in the last two decades due to its connection to several problems and particularly problems in data mining where frequently occurring patterns are of interest [3]. However, the question whether Trans-Hyp admits an output-polynomial time algorithm is still open. In fact, despite the number of papers on Trans-Hyp (see for instance these papers [3,13,14] and their bibliography section), the best known algorithm for Trans-Hyp is the one by Fredman and Khachiyan [17] which runs in time $O(n^{\log(n)})$ where n is the size of the hypergraph plus the number of minimal transversals.

Despite the link between Dom and Trans-Hyp and the importance of Dom in other areas such as building protocols in networks [24], to our knowledge, the only paper dealing with Dom is the one by Fomin et al. [15]. This paper, based on the Measure and Conquer technique from exact algorithms, gives an algorithm for Dom. However, its running time is $O(1.7159^n)$, where n is the number of vertices of the input graph. Hence, this algorithm is not an output-polynomial time one for Dom. It just informs us that the number of minimal dominating sets in a graph is upper bounded by $O(1.7159^n)$. Moreover, the algorithm does not use the fact that we deal with graphs, and uses instead the closed neighbourhood hypergraph. In this paper, we tackle Dom by restricting ourselves to some classes of graphs, as in the case of many output-polynomial time algorithms for Trans-Hyp.

Our contribution. After some preliminaries in Section 2, we recall some known output-polynomial time algorithms for Dom in Section 3. These results are of two types: those that come from meta-theorems in parameterized complexity theory and those that can be obtained from tractable cases of Trans-Hyp. In Section 4 we consider Dom in split graphs [5]. Split graphs are interesting for several reasons. In particular, it is a non trivial sub-class of chordal graphs where no output-polynomial time algorithm for Dom is known, and two important variants of Dom, namely total dominating sets and connected dominating sets coincide in split graphs. Section 5 is devoted to the extension of the result in Section 4 to other classes of graphs. For that we introduce a notion of maximal extension (a set of edges added to the graph) that keeps invariant the set of minimal dominating sets. We show that graphs that have chordal graphs as maximal extensions are exactly the one having split graphs as maximal extensions and derive a polynomial delay algorithm for chordal P_6-free graphs. We discuss in Section 6 our second goal consisting in studying the relationship between Trans-Hyp and Dom. We first show that the enumeration of the minimal total dominating sets is equivalent to Trans-Hyp and the enumeration of connected dominating sets is Trans-Hyp-Hard. Both are Trans-Hyp-complete when restricted to split graphs. Then, we show that the decision prolem associated to the enumeration of minimal dominating sets containing a set is co-NP-complete.

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. The set of natural integers is denoted by $\mathbb{N}$. The size of a set A is denoted $|A|$.

We refer to [10] for our graph terminology. A graph G is a pair $(V(G), E(G))$, where $V(G)$ is the set of vertices and $E(G) \subseteq V(G) \times V(G)$ is the set of edges. A graph G is said to be *undirected* if $(x, y) \in E(G)$ implies $(y, x) \in E(G)$, hence we can write xy (equivalently yx). In this paper graphs are simple, loop-free and undirected. We let $G[X]$, called the sub-graph of G induced by $X \subseteq V_G$, the graph $(X, E(G) \cap (X \times X))$. A graph G is said *chordal* if it has no induced cycle of length greater than or equal to 4.

For a graph G, we let $N_G(x)$ be the set of neighbours of x, *i.e.* the set $\{y \mid xy \in E(G)\}$, 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 $\bigcup\limits_{x \in X} N_G[x]$ and $N_G[X] \setminus X$.

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 *minimal* if for any $x \in D$, $D \setminus \{x\}$ is not a dominating set. 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 if $y \in N_G[x] \setminus N_G[D \setminus x]$. The set of private neighbours of x in D is denoted $P_D(x)$. The following is straightforward.

Lemma 1. *Let D be a minimal dominating set of a graph G. Then for all $x \in D$ we have $P_D(x) \neq \emptyset$.*

A *hypergraph* $\mathcal{H}$ is a pair $(V(\mathcal{H}), \mathcal{E}(\mathcal{H}))$ where $V(\mathcal{H})$ is a finite set and $\mathcal{E}(H) \subseteq 2^{V(\mathcal{H})}$. It is worth noticing that graphs are special cases of hypergraphs. By abuse of notations, we will call the elements of $\mathcal{E}(\mathcal{H})$ edges. A *transversal* (or *hitting set*) of $\mathcal{H}$ is a set $A \subseteq V$ that meets every edge of $\mathcal{E}(\mathcal{H})$. A transversal is *minimal* if it does not contain any other transversal as a subset. The set of all minimal transversals of $\mathcal{H}$ is denoted $Tr(\mathcal{H})$. The size of a hypergraph $\mathcal{H}$, denoted $||\mathcal{H}||$, is $|V(\mathcal{H})| + \sum\limits_{e \in \mathcal{E}(H)} |e|$.

Let $f : \mathbb{N} \to \mathbb{N}$. For a hypergraph $\mathcal{H}$ and $\mathscr{C} \subseteq 2^{V(\mathcal{H})}$, we say that an algorithm enumerates $\mathscr{C}$ with delay $f(||\mathcal{H}||)$ if, after a pre-processing that takes time $p(||\mathcal{H}||)$ for some polynomial p, it outputs the elements of $\mathscr{C}$ without repetitions, the delay between two outputs being bounded by $f(||\mathcal{H}||)$. If f is a polynomial, we call it a polynomial delay algorithm. We denote by TRANS-HYP, the enumeration problem of minimal transversals in hypergraphs. Similarly we denote by DOM, the enumeration problem of minimal dominating sets in graphs.

It is well-known that DOM can be polynomially reduced to TRANS-HYP as follows. 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 2. *[5] 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 $\mathcal{N}(G)$.*

Let us finish these preliminaries by some constructions of graphs from hypergraphs. If $\mathcal{H}$ is a hypergraph, we let $\mathcal{I}(\mathcal{H})$, 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 $\{xy_e \mid x \in V(G)$ and $x \in e\}$. $\mathcal{I}'(\mathcal{H})$, the *split incidence* graph of $\mathcal{H}$, is the graph obtained from $\mathcal{I}(\mathcal{H})$ by replacing $\mathcal{I}(\mathcal{H})[V(\mathcal{H})]$ by a clique on $V(\mathcal{H})$. Note that the neighbourhood of the vertex y_e in $\mathcal{I}(\mathcal{H})$ is exactly the set e. See Figure 1 for an example of $\mathcal{I}(\mathcal{H})$ and $\mathcal{I}'(\mathcal{H})$.

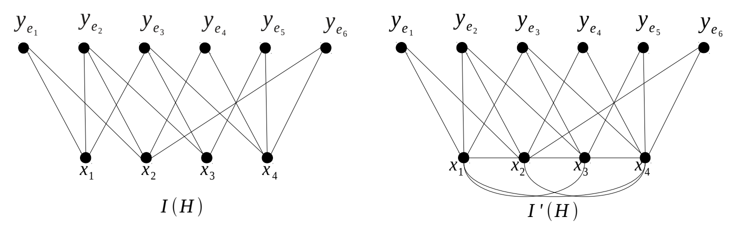

Fig. 1. An example of the bipartite incidence graph $\mathcal{I}(\mathcal{H})$ and the split incidence graph $\mathcal{I}'(\mathcal{H})$ of the hypergraph $\mathcal{H} = (\{x_1, x_2, x_3, x_4\}, \{e_1, e_2, e_3, e_4, e_5, e_6\})$ where $e_1 = \{x_1, x_2\}$, $e_2 = \{x_1, x_2, x_3\}$, $e_3 = \{x_1, x_3, x_4\}$, $e_4 = \{x_2, x_4\}$, $e_5 = \{x_3, x_4\}$, $e_6 = \{x_2, x_4\}$.

3 Examples of Tractable Cases

Even if DOM is not a well-studied problem, there are some known tractable cases. We review some of them.

Let us first start with the well-known graph classes in parameterized complexity theory. *Tree-width* [20] and *clique-width* [9] are well-known complexity measures in graph theory. The first is well-known thanks to its important role in the proof of the *Graph Minor Theorem* [21] and the notorious Courcelle's meta-theorem [6]. Courcelle's meta-theorem says that every decision and optimization graph problem expressible in *monadic second-order logic* can be solved in polynomial-time in graph classes of bounded tree-width. Most of the well-known NP-complete problems are expressible in monadic second-order logic, *e.g.* 3-colorability, computing the minimum dominating set, the minimum vertex-cover, etc. Deciding if a set is a minimal dominating set is also expressible in monadic second-order logic. Clique-width is important because it not only generalizes tree-width, but it also yields a meta-theorem (see [8]) similar to the one for tree-width. Clique-width generalizes tree-width in the sense that every class of graphs of bounded tree-width has bounded clique-width, but the converse is false, complete graphs have clique-width 2 and unbounded tree-width. A natural question is whether these meta-theorems [6,8] can be extended to counting and enumeration problems. In fact, the results in [6,8] are also stated for counting problems. And, in [7], Courcelle extends them to enumeration problems. He proved that if

$P(x_1,\ldots,x_m,X_1,\ldots,X_q)$ is a graph property, depending on vertices $x_1,\ldots,x_m$ and sets of vertices $X_1,\ldots,X_q$, expressible in monadic second-order logic and $\mathcal{C}$ is a class of graphs of bounded tree/clique-width then there exists an algorithm that for every graph G in $\mathcal{C}$ enumerates the set $\{(u_1,\ldots,u_m,Z_1,\ldots,Z_q) \mid G$ satisfies $P(u_1,\ldots,u_m,Z_1,\ldots,Z_q)\}$ with linear delay and uses linear space. The proof is rather involved and uses, as the other meta-theorems, machinery from logical tools. However, even if this "enumeration" meta-theorem is interesting and general, there are many natural graph classes that do not have bounded tree/clique-width, *e.g.* interval graphs, split graphs, planar graphs, etc. For some of them, one can prove that DOM is tractable by using a translation to tractable cases of TRANS-HYP. The following is an exemple.

Proposition 1. DOM *admits a polynomial delay algorithm when restricted to*

1. *Strongly chordal graphs.*
2. *Graph classes of bounded degeneracy.*

Proof. (1) If a graph G is strongly chordal, then $\mathcal{N}(G)$ is β-acyclic (see for instance [5]). By [11], TRANS-HYP admits a polynomial delay algorithm in β-acyclic hypergraphs.

(2) In [13], it is defined a notion of degeneracy for hypergraphs that extends the one on graphs. One easily verifies that if G is k-degenerate, then so is $\mathcal{N}(G)$. Since TRANS-HYP admits a polynomial delay algorithm in degenerate hypergraphs [13], we are done. □

Examples of strongly chordal graphs are directed path graphs (which include interval graphs), comparability chordal graphs, etc. Examples of degenerate graph classes are planar graphs, bounded-degree graphs, graphs of bounded genus, graphs of bounded tree-width, etc. Hence, the tractability of DOM in graph classes of tree-width k can be derived from Proposition 1(2). However, the use of Courcelle's result yields better bounds. By using [7], we have that DOM admits an algorithm with delay $f(k)\cdot n$ for some function $f:\mathbb{N}\to\mathbb{N}$, while [13] proves that DOM admits an algorithm with delay $n^{g(k)}$ for some function $g:\mathbb{N}\to\mathbb{N}$.

The notion of tree-width was extended to hypergraphs. However, there exist several notions (see for instance [1]). If we define the tree-width of a hypergraph $\mathcal{H}$ as the tree-width of its incidence graph $\mathcal{I}(\mathcal{H})$, then TRANS-HYP admits a polynomial delay algorithm in hypergraphs of bounded tree-width [13]. But, for the other notions, TRANS-HYP is hard even for classes of simple hypergraphs with *hypertree-width* 2 [12]. A natural question is whether there exists a class $\mathcal{C}$ of graphs with unbounded tree-width but such that $\mathcal{N}(\mathcal{C}) := \{\mathcal{N}(G) \mid G \in \mathcal{C}\}$ has bounded tree-width. The following proposition proves that it is not possible ($twd(G)$ denotes the tree-width of a graph G). Its proof (as most of the proofs in this extended abstract) is omitted because of space constraints.

Proposition 2. *For every graph G,*

$$\frac{twd(\mathcal{I}(\mathcal{N}(G)))-1}{2} \leq twd(G) \leq twd(\mathcal{I}(\mathcal{N}(G))).$$

4 Dom 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 maximal.) Hence, we will denote a split graph G by the pair $(C(G) \cup S(G), E(G))$. We prove in this section that Dom in split graphs admits a linear delay algorithm that uses polynomial space. We first notice that a minimal dominating set D in 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 3 shows that a minimal dominating set D is characterized by D_C. Note that D_S cannot characterize D_C, since several minimal dominating sets can have the same set D_S.

Lemma 3. *Let G be a split graph and D a minimal dominating set of G. Then $D_S = S \setminus N_G(D_C)$.*

Lemma 4. *Let G be a split graph and D be a minimal dominating set of G. Then for all $A \subseteq D_C$, the set $A \cup (S(G) \setminus N_G(A))$ is a minimal dominating set of G.*

Proof. Let D be a minimal dominating set of G and $A \subseteq D_C$. We show that $D' = A \cup (S(G) \setminus N_G(A))$ is a minimal dominating set. If $A = \emptyset$, then $D' = S(G)$ and since $S(G)$ is a maximal independent set, it is a minimal dominating set. Now suppose that $A \neq \emptyset$ and $x \in A$. Clearly $P_D(x) \neq \emptyset$ since D is minimal. This implies that $P_{D'}(x)$ is also not empty. Moreover, for any element $y \in D'_S$, we have $P_{D'}(y) = \{y\}$. We conclude that D' is a minimal dominating set. □

A consequence of Lemmas 3 and 4 is the following.

Corollary 1. *Let G be a split graph. Then, there is a bijection between $\mathcal{D}(G)$ and the set $\mathcal{DI}(G) = \{D_C \mid D \in \mathcal{D}(G)\}$. The set $\mathcal{DI}(G)$ is moreover closed under inclusion (*i.e.*, is an independent system).*

So the generation of minimal dominating sets of a split graph is equivalent to the generation of elements in $\mathcal{DI}(G)$. In the following we give a linear delay algorithm to generate $\mathcal{DI}(G)$. Let $D, D' \in \mathcal{DI}(G)$. We say that D' covers D if $D \subseteq D'$ and $D' \setminus D$ is a singleton. We denote by $COV(D)$ the set $\{x \in C \setminus D \mid D \cup \{x\} \text{ covers } D\}$. If $D \in \mathcal{DI}(G)$ then y in $C \setminus D$ belongs to $COV(D)$ if each vertex in $D \cup \{y\}$ has a private neighbour. In order to enumerate $\mathcal{DI}(G)$, we call Dominant$(\emptyset, C(G))$.

Theorem 1. *Algorithm 1 generates the set $\mathcal{DI}(G)$ with $O(m+n)$ delay and uses space bounded by $O(n^2)$.*

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 edges, *w.r.t.* inclusion, in $\mathcal{N}(G)$.

Algorithm 1: Dominant(D,COV)

Data : a split graph G
Result : $\mathcal{DI}(G)$
begin
    Output(D)
    foreach $x \in COV$ **do**
        COV=COV$\setminus\{x\}$
        NewCOV= $\emptyset$
        foreach $y \in COV$ **do**
            if *each vertex in* $D \cup \{x, y\}$ *has a private neighbour* **then**
                NewCOV=NewCOV$\cup\{y\}$
        Dominant($D \cup \{x\}$,NewCOV)
end

Let G be a graph. A vertex $x \in V(G)$ is said to be *irredundant* if for all $y \neq x$, $N_G[y] \not\subseteq N_G[x]$, otherwise it is called *redundant*. In case of twins, we choose arbitrarily one to being irredundant, the others are so redundant. The set of irredundant (resp. redundant) vertices is denoted by $M(G)$ (resp. $RN(G)$). 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 replacing $G[RN(G)]$ by a clique on $RN(G)$. Note that the completion graph of a split graph G is G itself. 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 2 gives some examples of completion graphs.

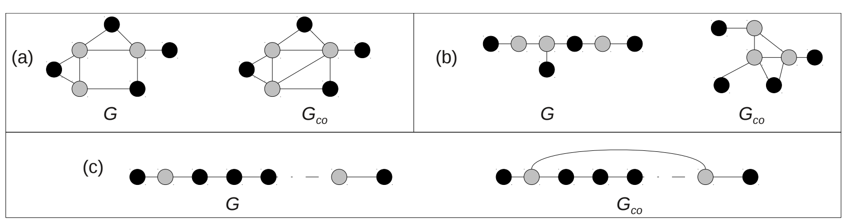

Fig. 2. (a) a non-chordal graph, its completion is a split graph (b) a chordal graph with an induced P_6, its completion is a split graph (c) a path P_n, its completion is not chordal.

Lemma 5. *For any graph G, we have $\mathcal{D}(G) = \mathcal{D}(G_{co})$.*

In the following we are interested in graphs such that Dom in their completion graphs has an efficient generation algorithm. For two graphs G and H, we say that G is H-free if it does not contain H as an induced subgraph. For $k \geq 1$, we let P_k be the path on k vertices. A vertex is simplicial if the graph induced by its neighbourhood is a clique.

Lemma 6. *If G is a P_6-free chordal graph, then for all $x \in M(G)$, x is a simplicial vertex in G_{co}. Furthermore, the set $M(G)$ is an independent set in G_{co}.*

Proposition 3. *Let G be a P_6-free chordal graph. Then G_{co} is a split graph.*

Proof. From Lemma 6, it follows that $M(G)$ forms an independent set in G_{co}, and since $RN(G)$ forms a clique, we are done. □

The next theorem characterizes completion graphs that are split.

Theorem 2. *Let G be a graph. Then G_{co} is a chordal graph if and only if G_{co} is a split graph.*

6 Related Problems

In this section we discuss some variants of dominating sets and related problems. The enumeration of total dominating sets polynomially reduces to Trans-Hyp with respect to the classical Karp reduction. However, another natural variant, the enumeration of connected dominating sets is not known to have a reduction to Trans-Hyp, we show here that it is harder than Trans-Hyp. The problem in Section 6.3 is motivated by the investigation of an enumeration algorithm for dominating sets, which is inspired from an approach in [4].

6.1 Total Dominating Set

A total dominating set can be viewed as a dominating set in which the vertices do not cover themselves. A *total dominating set* of a graph G is a subset of vertices $D \subseteq V(G)$ such that for all $x \in V(G)$, $N_G(x) \cap D \neq \emptyset$; D is minimal if for all $x \in D$, $D \setminus \{x\}$ is not a total dominating set. We note $\mathcal{D}_t(G)$ the set of all minimal total dominating sets of G. For a graph G, we let $\mathcal{N}_o(G)$, the *open neighbourhood hypergraph* be $(V(G), \{N_G(x) \mid x \in V(G)\})$. We let Tds be the problem of listing $\mathcal{D}_t(G)$ for a graph G.

Proposition 4. Tds *is equivalent to* Trans-Hyp*.*

Proof. We first show that one can reduce Tds to Trans-Hyp on open neighbourhood hypergraph (the reduction was first noted in [22]). Let G be a graph. It is enough to show that $D \subseteq V(G)$ is a total dominating set in G if and only if it is a transversal of $\mathcal{N}_o(G)$. 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 show now that Trans-Hyp can be reduced to Tds. 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 $Tr(\mathcal{H}) = \{\{x\}\} \cup Tr(\mathcal{H} \setminus \{x\})$ and consider so this reduced hypergraph. We then show that $\mathcal{D}_t(\mathcal{I}'(\mathcal{H})) = Tr(\mathcal{H})$.

(i) Let D be a minimal total dominating set of $\mathcal{I}'(\mathcal{H})$ and let $e \in \mathcal{E}(\mathcal{H})$. Then, there exists $x \in V(\mathcal{H}) \cap D$ such that $xy_e \in E(\mathcal{I}'(\mathcal{H}))$, *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 $\mathcal{I}'(\mathcal{H})[V(\mathcal{H})]$ is a clique, $D \setminus y_e$ is also a total dominating set ($D \cap V(\mathcal{H}) \geq 2$), 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 $z \in V(\mathcal{I}'(\mathcal{H})) \setminus V(\mathcal{H})$ there exists $x \in T$ such that $xz \in E(\mathcal{I}'(\mathcal{H}))$. Since there is no dominating vertex, $|T| \geq 2$, and because $\mathcal{I}'(\mathcal{H})[V(\mathcal{H})]$ is a clique, for all $x \in V(\mathcal{H})$, there exists $y \in T$ such that $xy \in E(\mathcal{I}'(\mathcal{H}))$. Hence, T is a total dominating set of $\mathcal{I}'(\mathcal{H})$.

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

Remark 1. The proof of Proposition 4 reveals that Trans-Hyp is reduced to Tds in split graphs. Hence, Tds in graphs is equivalent to Tds in split graphs.

6.2 Connected Dominating Set

A *connected dominating set* in a graph G is a subset D of $V(G)$ such that D is a dominating set of G and such that $G[D]$ is connected. A connected dominating set D is minimal, if for all $x \in D$, $D \setminus \{x\}$ is not a connected dominating set, in other words either $D \setminus \{x\}$ is not a dominating set or $G[D \setminus \{x\}]$ is not connected. We denote by $\mathcal{D}_c(G)$ the set of all minimal connected dominating sets of G. We let Cds the problem of generating $\mathcal{D}_c(G)$ for a graph G.

Proposition 5. Cds *in split graphs is equivalent to* Trans-Hyp.

Proof. Let $\mathcal{H}$ be a hypergraph. Then we show that $\mathcal{D}_c(\mathcal{I}'(\mathcal{H})) = Tr(\mathcal{H})$.

(i) Let $D \in \mathcal{D}_c(\mathcal{I}'(\mathcal{H}))$. Note firstly that $D \subseteq V(\mathcal{H})$. Indeed, suppose that there is $y_e \in D$ for some $e \in \mathcal{E}(\mathcal{H})$. Since, D must be connected, there is a neighbour z of y_e in D. Since $\{y_e \mid e \in \mathcal{E}(\mathcal{H})\}$ is an independent set, z must belong to $V(\mathcal{H})$. But since $\mathcal{I}'(\mathcal{H})[V(\mathcal{H})]$ forms a clique, $P_D(y_e) \subseteq P_D(z)$ and thus $D \setminus \{y_e\}$ is yet a connected dominating set, which contradicts the minimality of D. Now, for each $e \in \mathcal{E}(\mathcal{H})$, there exists $x \in D$ such that $xy_e \in E(\mathcal{I}'(\mathcal{H}))$, 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 $\mathcal{I}'(\mathcal{H})[V(\mathcal{H})]$ is a clique, T is connected and, for each $x \in V(\mathcal{H})$, there exists $y \in T$ such that $xy \in E(\mathcal{I}'(\mathcal{H}))$. Furthermore, for each $e \in \mathcal{E}(\mathcal{H})$, $T \cap e \neq \emptyset$, *i.e.* for each $y_e \in V(\mathcal{I}'(\mathcal{H})) \setminus V(\mathcal{H})$, there is $z \in T$ such that $zy_e \in E(\mathcal{I}'(\mathcal{H}))$. Hence, T is a connected dominating set of $\mathcal{I}'(\mathcal{H})$.

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

It remains to reduce Cds to Trans-Hyp. 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 $G = \mathcal{I}'(\mathcal{H})$ and so from above, $\mathcal{D}_c(\mathcal{I}'(\mathcal{H})) = Tr(\mathcal{H})$. □

Remark 2. We do not currently know if CDS is equivalent to TRANS-HYP in all graphs. However, CDS in bipartite graphs is TRANS-HYP-Hard. Indeed consider, for a hypergraph $\mathcal{H}$, the graph B defined as follows: $V(B) = V(\mathcal{I}(\mathcal{H})) \cup \{x, y\}$ and $E(B) = E(\mathcal{I}(\mathcal{H})) \cup \{xy\} \cup \{xz \mid z \in V(\mathcal{H})\}$, then one can easily show that $Tr(\mathcal{H}) = \mathcal{D}_c(B)$.

We can remark that $\mathcal{D}_c(G)$ and $\mathcal{D}_t(G)$ are equal in split graphs, and they also coincide with another set $\mathcal{D}_{mc}(G) := \mathcal{D}_c(G) \cap \mathcal{D}(G)$ which is the minimal dominating sets that are connected. Note that $\mathcal{D}_{mc}(G)$ can be empty in general.

6.3 Dominating Sets Containing a Set

For a hypergraph $\mathcal{H}$ and a subset A of $V(\mathcal{H})$, we denote by $Tr(\mathcal{H}, A)$, the set of minimal transversals of $\mathcal{H}$ containing A. The problem consisting in asking whether $\mathcal{T} = Tr(\mathcal{G}, A)$, given $\mathcal{T} \subseteq 2^{V(\mathcal{H})}$, is denoted by TCS.

Proposition 6. *[4]* TCS *is co-NP-complete.*

For a graph G and a subset A of $V(G)$, we denote by $\mathcal{D}(G, A)$, the set of minimal dominating sets containing A. The problem consisting in asking whether $\mathcal{T} = \mathcal{D}(G, A)$, given $\mathcal{T} \subseteq 2^{V(G)}$, is denoted by DCS.

Proposition 7. DCS *is co-NP-complete.*

Proof. DCS is in coNP. It suffices to guess a set of vertices and check in polynomial time if this set is a dominating set containing A and not in $\mathcal{T}$. So it is sufficient to show the reduction from TCS to DCS. Let $\mathcal{H}$ be a hypergraph and $A \subseteq V(\mathcal{H})$. We construct the graph B such that $V(B) = V(\mathcal{I}'(\mathcal{H})) \cup \{w, x, y, z\}$, and $E(B) = E(\mathcal{I}'(\mathcal{H})) \cup \{wx, xy, yz\} \cup \{\{wy_e\} \mid e \in \mathcal{E}(\mathcal{H})\}$. An example is given in Figure 3. We show that $Tr(\mathcal{H}, A)$ is in bijection with $\mathcal{D}(B, A \cup \{x, y\})$.

(i) Let $T \in Tr(\mathcal{H}, A)$ then we claim that $T' = T \cup \{x, y\} \in \mathcal{D}(B, A \cup \{x, y\})$. Indeed $A \cup \{x, y\} \subseteq T'$ and all vertices in $V(\mathcal{H})$ are covered because $\mathcal{I}'(\mathcal{H})[V(\mathcal{H})]$ forms a clique and $T \neq \emptyset$. Furthermore for all $e \in \mathcal{E}\mathcal{H}$, $e \cap T \neq \emptyset$, so $N_B[y_e] \cap T' \neq \emptyset$ and w and z are covered by T' because $\{x, y\} \subseteq T'$. We must also check that T' is minimal. Since $T \in Tr(\mathcal{H}, A)$, it is clear that we can not remove a vertex in T, and we can not remove neither x nor y otherwise, w or z would not be covered. So $T' \in \mathcal{D}(B, A \cup \{x, y\})$.

(ii) Let now $D \in \mathcal{D}(B, A \cup \{x, y\})$, we show that $D' = D \setminus \{x, y\} \in Tr(\mathcal{H}, A)$. We first claim that $D' \subseteq V(\mathcal{H})$. Actually, since D is a minimal dominating set, for all $z \in D$, $P_D(z) \neq \emptyset$. But $P_D(x) \subseteq \{w\}$ and so, if y_e, for some $e \in \mathcal{E}(\mathcal{H})$, belong to T, then $P_D(x)$ would be empty, which contradicts the minimality of D. Also, w and z cannot belong to D, otherwise $P_D(x)$ or $P_D(y)$ would be empty. Furthermore, since for all $e \in \mathcal{E}(\mathcal{H})$, y_e must be covered by some vertex in D and since $D \subseteq V(\mathcal{H})$, D' is a transversal of $\mathcal{H}$. Finally, by definition, $A \subseteq D'$, and then D' is a transversal of $\mathcal{H}$ containing A.

From (i) and (ii) we can conclude that $Tr(\mathcal{H}, A) = \{D \setminus \{x, y\} \mid D \in \mathcal{D}(B, A \cup \{x, y\})\}$ and hence TCS is reduced to DCS. □

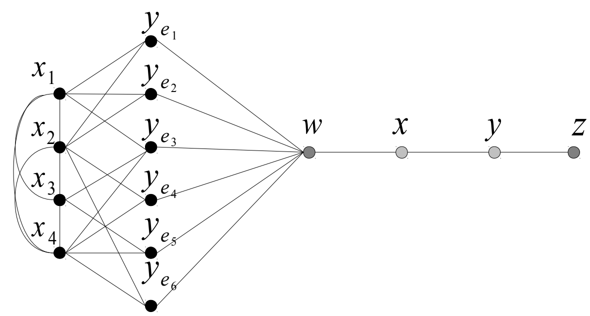

Fig. 3. An example of $B(\mathcal{H})$ where $\mathcal{H} = (\{x_1, x_2, x_3, x_4\}, \{e_1, e_2, e_3, e_4, e_5, e_6\})$ with $e_1 = \{x_1, x_2\}$, $e_2 = \{x_1, x_2\}$, $e_3 = \{x_1, x_3, x_4\}$, $e_4 = \{x_2, x_4\}$, $e_5 = \{x_3, x_4\}$, $e_6 = \{x_2, x_4\}$.

ACKNOWLEDGMENTS. We would like to thank the anonymous referees for their helpful suggestions and comments. A. Mary and L. Nourine are partially supported by the ANR (french National Research Agency) project DAG (ANR-09-DEFIS, 2009-2012). M.M. Kanté and V. Limouzy are supported by the ANR junior project DORSO (2011-2015).

References

1. Adler, I., Gottlob, G., Grohe, M.: Hypertree-Width and Related Hypergraph Invariants. European Journal of Combinatorics 28, 2167–2181 (2007)
2. Agrawal, R., Imielinski, T., Swami, A.N.: Database Mining: A Performance Perspective. IEEE Trans. Knowl. Data Eng. 5(6): 914-925 (1993)
3. Boros, E., Elbassioni, K., Khachiyan, L., Gurvich., V.: 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.
4. Boros, E., Gurvich, V., Hammer, P.L.: Dual subimplicants of positive Boolean functions, Optimization Methods and Software, 10 (1998) pp. 147-156.
5. Brandstädt, A., Bang Le, V., Spinrad, J.P.: Graph Classes a Survey. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia (1999)
6. Courcelle, B.: The Monadic Second-Order Logic of Graphs I: Recognizable Sets of Finite Graphs. Inf. Comput. 85(1), 12–75 (1990)
7. Courcelle, B.: Linear Delay Enumeration and Monadic Second-Order Logic. Discrete Applied Mathematics, 157, 2675–2700 (2009)
8. Courcelle, B., Makowsky, J. A., Rotics, U.: Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. Theory of Computing Systems 33(2), 125–150 (2000)
9. Courcelle, B., Olariu, S.: Upper Bounds to the Clique-Width of Graphs. Discrete Applied Mathematics 101(1-3), 77–114 (2000)
10. Diestel, R.: Graph Theory. Springer-Verlag, 3^{rd} edition (2005)
11. Eiter, T., Gottlob, G.: Identifying the Minimal Transversals of a Hypergraph and Related Problems. SIAM Journal on Computing 24(6), 1278–1304 (1995)

12. Eiter, T., Gottlob, G.: Hypergraph Transversal Computation and Related Problems in Logic and AI. In Flesca, S., Greco, S., Leone, S., Giovambattista, I. (eds) JELIA 2002. LNCS, vol. 2424, pp 549–564. Springer, (2002)
13. Eiter, T., Gottlob, G., Makino, K.: New Results on Monotone Dualization and Generating Hypergraph Transversals. SIAM Journal on Computing 32(2), 514–537 (2003)
14. Eiter, T., Makino, K., Gottlob, G.: Computational aspects of monotone dualization: A brief survey. Discrete Applied Mathematics 156(11): 2035-2049 (2008)
15. Fomin, F.V., Grandoni, F., Pyatkin, A., Stepanov,A.: Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. ACM Transactions on Algorithms, 5(1): 2008.
16. Kanté, M.M., Limouzy, V., Mary, A., Nourine, L.: On the Enumeration of Minimal Dominating Sets and Related Notions. Manuscript (2011)
17. Fredman, M., Khachiyan, L.: On the complexity of dualization of monotone disjunctive normal forms. Journal of Algorithms, 21(3):618–628, 1996.
18. Gély, A., Nourine, L., Sadi, B.: Enumeration aspects of maximal cliques and bicliques, Discrete Applied Mathematics 157(7): 1447-1459 (2009)
19. Johnson, D. S., Yannakakis, M., Papadimitriou, C. H.:On generating all maximal independent sets. Information Processing Letters, 27, 119–123(1988).
20. Robertson, N., Seymour, P.D.: Graph minors V:Excluding a Planar Graph. J. Comb. Theory, Ser. B 41, 92–114 (1986)
21. Robertson, N., Seymour, P.D.: Graph minors XX:Wagner's Conjecture. J. Comb. Theory, Ser. B 92(2), 325-357 (2004)
22. Thomassé, S., Yeo, A.: Total domination of graphs and small transversals of hypergraphs. Combinatorica 27(4): 473-487 (2007)
23. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge: Cambridge University Press. (1994)
24. Wu, J., Li, H.: A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks. Telecommunication Systems 18(1-3): 13-36 (2001)