



**HAL**  
open science

# Minimal dominating sets enumeration with FPT-delay parameterized by the degeneracy and maximum degree

Valentin Bartier, Oscar Defrain, Fionn Mc Inerney

► **To cite this version:**

Valentin Bartier, Oscar Defrain, Fionn Mc Inerney. Minimal dominating sets enumeration with FPT-delay parameterized by the degeneracy and maximum degree. 2023. hal-04162863

**HAL Id: hal-04162863**

**<https://hal.science/hal-04162863v1>**

Preprint submitted on 16 Jul 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Minimal dominating sets enumeration with FPT-delay parameterized by the degeneracy and maximum degree\*

Valentin Bartier<sup>†</sup>    Oscar Defrain<sup>‡</sup>    Fionn Mc Inerney<sup>§</sup>

May 12, 2023

## Abstract

At STOC 2002, Eiter, Gottlob, and Makino presented a technique called *ordered generation* that yields an  $n^{O(d)}$ -delay algorithm listing all minimal transversals of an  $n$ -vertex hypergraph of degeneracy  $d$ , for an appropriate definition of degeneracy. Recently at IWOCA 2019, Conte, Kanté, Marino, and Uno asked whether, even for a more restrictive notion of degeneracy, this XP-delay algorithm parameterized by  $d$  could be made FPT-delay parameterized by  $d$  and the maximum degree  $\Delta$ , *i.e.*, an algorithm with delay  $f(d, \Delta) \cdot n^{O(1)}$  for some computable function  $f$ . We answer this question in the affirmative whenever the hypergraph corresponds to the closed neighborhoods of a graph, *i.e.*, we show that the intimately related problem of enumerating minimal dominating sets in graphs admits an FPT-delay algorithm parameterized by the degeneracy and the maximum degree.

**Keywords:** algorithmic enumeration, hypergraph dualization, minimal transversals, minimal dominating sets, FPT-delay enumeration.

---

\*This work was supported by the ANR project DISTANCIA (ANR-17-CE40-0015) and the Austrian Science Foundation (FWF, project Y1329).

<sup>†</sup>Univ Lyon, CNRS, INSA Lyon, UCBL, Centrale Lyon, Univ Lyon 2, LIRIS, UMR5205, F-69622 Villeurbanne, France

<sup>‡</sup>Aix-Marseille Université, CNRS, LIS, 13009 Marseille, France

<sup>§</sup>Algorithms and Complexity Group, TU Wien, Austria

# 1 Introduction

Graphs and hypergraphs are ubiquitous in discrete mathematics and theoretical computer science. Formally, a *hypergraph*  $\mathcal{H}$  consists of a family  $E(\mathcal{H})$  of subsets called *edges* (or *hyperedges*) over a set  $V(\mathcal{H})$  of elements called *vertices*, and it is referred to as a *graph* when each of its edges has size precisely two. A *transversal* in a hypergraph is a set of vertices that intersects every edge. The problem of enumerating the family  $Tr(\mathcal{H})$  of all (inclusion-wise) minimal transversals of a given hypergraph  $\mathcal{H}$ , usually denoted by TRANS-ENUM and also known as *hypergraph dualization*, is encountered in many areas of computer science including logic, database theory, and artificial intelligence [KPS93, EG95, GMKT97, EMG08, NP12]. The best known algorithm to date for the problem is due to Fredman and Khachiyan [FK96], and runs in time  $N^{o(\log N)}$ , where  $N = |V(\mathcal{H})| + |E(\mathcal{H})| + |Tr(\mathcal{H})|$ . The existence of an output-polynomial-time algorithm is now open for more than 30 years,<sup>1</sup> and is arguably one of the most important problems in algorithmic enumeration [JYP88, EMG08]. An enumeration algorithm is said to be running in *output-polynomial* time if it outputs all solutions and stops in a time that is bounded by a polynomial in the combined sizes of the input and the output. It is said to be running with *polynomial delay* if the running times before the first output, after the last output, and between two consecutive outputs are bounded by a polynomial depending on the size of the input only. Clearly, any polynomial-delay algorithm defines an output-polynomial time algorithm. We refer the reader to [JYP88, Str19] for more details on the usual complexity measures of enumeration algorithms, and to [EMG08] for a survey on minimal transversals enumeration.

In 2002, an important step toward the characterization of tractable cases of TRANS-ENUM was made by Eiter, Gottlob, and Makino [EGM02]. The problem, studied in its equivalent formulation of monotone Boolean dualization, was shown to admit a polynomial-delay algorithm whenever the hypergraph has bounded degeneracy, for an appropriate notion of hypergraph degeneracy (see Section 2 for definitions of graph and hypergraph degeneracies) that generalizes graph degeneracy. Specifically, the authors proved the following result.

**Theorem 1.1** ([EGM02, EGM03]). *There is an algorithm that, given an  $n$ -vertex hypergraph of weak degeneracy  $d$ , generates its minimal transversals with  $n^{O(d)}$ -delay.*

In particular, the above theorem captures  $\alpha$ -acyclic hypergraphs and several other classes of hypergraphs related to restricted cases of conjunctive normal form (CNF)

---

<sup>1</sup>It should be noted that an output-polynomial time algorithm was recently claimed by Wild in an arXiv preprint [Wil23]; the proof, however, contains a major flaw (Claim (4), Section 3) that, to the best of our knowledge, has not been corrected since.

formulas. Another consequence of Theorem 1.1 is a polynomial-delay algorithm for the intimately related problem of enumerating minimal dominating sets in graphs, usually denoted by DOM-ENUM, when the graph has bounded degeneracy. A *minimal dominating set* in a graph is a (inclusion-wise) minimal subset of vertices intersecting each closed neighborhood. As the family of closed neighborhoods of a graph defines a hypergraph, DOM-ENUM naturally reduces to TRANS-ENUM. However, quite surprisingly, and despite the fact that not all hypergraphs are hypergraphs of closed neighborhoods of a graph, the two problems were actually shown to be equivalent in [KLMN14]. This led to a significant interest in DOM-ENUM, and the characterization of its complexity status has since been explored in various graph classes in the literature [KLM<sup>+</sup>15, GHK<sup>+</sup>16, GHK<sup>+</sup>18, BDH<sup>+</sup>20].

The theory of parameterized complexity provides a framework to analyze the running time of an algorithm as a function of the input size and some parameters of the input such as degeneracy. It has proved to be very useful in finding tractable instances of NP-hard problems, as well as for devising practical algorithms. In this context, a problem is considered tractable for a fixed parameter  $k$  if it is FPT *parameterized by  $k$* , that is, if it admits an algorithm running in time  $f(k) \cdot n^{O(1)}$  for some computable function  $f$ , where  $n$  is the size of the input. It is important to note that the combinatorial explosion of the problem can be restrained to the parameter  $k$  here. There are also weaker notions of tractability in this context, with a problem being XP *parameterized by  $k$*  (for some parameter  $k$ ) if it admits an algorithm running in time  $n^{f(k)}$  for some computable function  $f$ , where  $n$  is again the size of the input. Studying the tractability of important problems within the paradigm of parameterized complexity has led to a fruitful line of research these last decades, and we refer the reader to [CFK<sup>+</sup>15] for a relatively recent comprehensive overview of techniques and results in this field.

The study of algorithmic enumeration from a parameterized complexity point of view is quite novel. It may be traced back to works such as [Fer02, Dam06], where the authors devise algorithms listing all solutions of size  $k$  in time  $f(k) \cdot n^{O(1)}$ , where  $n$  is the size of the input. Note, however, that these kinds of results hint at a relatively low number of solutions (typically not superpolynomial in  $n$  for fixed  $k$ ), while this is not the case for many enumeration problems including the one that we consider in this paper. New parameterized paradigms for enumeration problems of this second type were later developed in [CMM<sup>+</sup>17, GKKL22], and a growing interest for FPT-enumeration has since been developing as witnessed by the recent Dagstuhl Seminar on output-sensitive, input-sensitive, parameterized, and approximative enumeration [FGS19], and the recent progress on this topic [Mei18, CKMU19, CKM<sup>+</sup>19, Mei20, GKKL22]. Among the different complexity measures

one could want to satisfy, the notion of FPT-delay naturally arises. An algorithm  $A$  for an enumeration problem parameterized by  $k$  runs with FPT-*delay* if there exists a computable function  $f$  such that  $A$  has delay  $f(k) \cdot n^{O(1)}$ , where  $n$  is the size of the input. An XP-*delay* algorithm for an enumeration problem parameterized by  $k$  is analogously defined with the delay being  $n^{f(k)}$  instead. Note that the algorithm given by Theorem 1.1 is XP-*delay*, but not FPT-*delay*. Recently at IWOCA 2019, Conte, Kanté, Marino, and Uno asked whether, even for a more restrictive notion of degeneracy that is defined in Section 2, this algorithm could be improved to run with FPT-*delay* parameterized by both the degeneracy  $d$  and the maximum degree  $\Delta$ , *i.e.*, the maximum number of hyperedges a vertex may intersect. More precisely, they stated the following question.

**Question 1.2** ([CKMU19]). *Can the minimal transversals of an  $n$ -vertex hypergraph be listed with delay  $2^d \cdot \Delta^{f(d)} \cdot n^{O(1)}$ , for some computable function  $f$ , where  $d$  is the strong degeneracy of the hypergraph and  $\Delta$  its maximum degree?*

The given time bound in the above question is not arbitrary. It is chosen to match the complexity of an FPT-*delay* algorithm the authors in [CKMU19] give for another related problem, namely the enumeration of maximal irredundant sets in hypergraphs. A set of vertices is *irredundant* if removing any of its vertices decreases the number of edges it intersects. It is easy to see that minimal transversals are maximal irredundant sets. However, the converse is not true, and the two problems of listing minimal transversals and maximal irredundant sets behave differently. Indeed, while TRANS-ENUM is known to admit an output quasi-polynomial-time algorithm [FK96], the enumeration of maximal irredundant sets is known to be impossible in output-polynomial time unless  $P = NP$  [BM16]. As put in [CKMU19], answering Question 1.2 would help precisizing the links between the two problems, as maximal irredundancy is commonly believed to be more difficult than minimal transversality (and domination). In the same paper, the authors note that their algorithm for maximal irredundancy does not apply to minimal transversality, even in the more restricted but open case where the hypergraph consists of the closed neighborhoods of some graph, *i.e.*, in the context of graph domination. The next question naturally follows, and may be regarded as a weakening of Question 1.2 that is implicit in [CKMU19].

**Question 1.3** ([CKMU19]). *Can the minimal dominating sets of an  $n$ -vertex graph be listed with delay  $2^d \cdot \Delta^{f(d)} \cdot n^{O(1)}$ , for some computable function  $f$ , where  $d$  is the degeneracy of the graph and  $\Delta$  its maximum degree?*

This paper is devoted to giving a positive answer to Question 1.3, suggesting that minimal domination and maximal irredundancy are of relatively comparable

tractability as far as the degeneracy is concerned. Not only do we answer Question 1.3 in the affirmative, but we prove a slightly stronger result. That is, that an FPT-delay algorithm for TRANS-ENUM exists when parameterized by the degeneracy and the maximum size of a hyperedge, usually referred to as *dimension*, even for the notion of hypergraph degeneracy considered in [EGM02, EGM03], which is less restrictive than the one considered in [CKMU19], a point that is discussed in Section 2. More formally, we prove the following result.

**Theorem 1.4.** *There is a computable function  $f$  such that the minimal transversals of an  $n$ -vertex hypergraph can be enumerated with delay  $2^d \cdot k^{f(d)} \cdot n^{O(1)}$ , where  $d$  is the weak degeneracy of the hypergraph and  $k$  is its dimension.*

The algorithm we obtain is based on the ordered generation technique originally presented by Eiter *et al.* in [EGM02], combined with the use of a bounded number of equivalence classes with respect to the edges that can be hit by a well-chosen vertex of bounded “backdegree”. While the techniques of this paper are not novel, we believe that our presentation is of valuable interest as far as the ordered generation technique is concerned. Furthermore, it is a preliminary step toward the resolution of Question 1.2, and makes headway in the recent and promising line of research concerning the study of enumeration problems in the parameterized setting.

**Organization of the paper.** In the next section, we introduce the problem and the related notations, and discuss the aforementioned variants of hypergraph degeneracy considered in [EGM03, CKMU19]. In Section 3, we present the ordered generation technique applied to hypergraphs, and show how it amounts to generating extensions of a partial solution while preserving FPT-delay. In Section 4, we show how to generate extensions with FPT-delay parameterized by the degeneracy and dimension, proving Theorem 1.4. The answer to Question 1.3 is given in Section 5 as a corollary. Future research directions and the limitations of the presented techniques are discussed in Section 6.

## 2 Preliminaries

Let  $\mathbb{N}^*$  denote the set of positive integers.

**Graphs and hypergraphs.** Recall that, for a hypergraph  $\mathcal{H}$ , its vertex set is denoted by  $V(\mathcal{H})$ , and its edge set is denoted by  $E(\mathcal{H})$ . If every edge has size precisely two, then it is referred to as a graph and is usually denoted by  $G$ . Given a

vertex  $v \in V(\mathcal{H})$ , the set of edges that contain  $v$  (also called edges *incident* to  $v$ ) is denoted by  $\text{inc}(v)$ . The *degree* of  $v$  in  $\mathcal{H}$ , denoted by  $\text{deg}_{\mathcal{H}}(v)$ , is the number of edges in  $\text{inc}(v)$ . The maximum degree of a vertex in  $\mathcal{H}$  is denoted by  $\Delta(\mathcal{H})$  (or simply  $\Delta$ ). The *dimension* of  $\mathcal{H}$  is the maximum size of an edge in  $\mathcal{H}$ . A *transversal* of  $\mathcal{H}$  is a subset  $T \subseteq V(\mathcal{H})$  such that  $E \cap T \neq \emptyset$  for all  $E \in E(\mathcal{H})$ . It is called *minimal* if it is minimal by inclusion, or equivalently, if  $T \setminus \{v\}$  is not a transversal for any  $v \in T$ . The set of minimal transversals of  $\mathcal{H}$  is denoted by  $\text{Tr}(\mathcal{H})$ .

In this paper, we are interested in the following problem known to admit an XP-delay algorithm [EGM03] parameterized by the degeneracy, but for which the existence of an FPT-delay algorithm parameterized by the same parameter (even in addition to the maximum degree) is open [CKMU19].

MINIMAL TRANSVERSALS ENUMERATION (TRANS-ENUM)

**Input:** A hypergraph  $\mathcal{H}$ .

**Output:** The set  $\text{Tr}(\mathcal{H})$ .

We introduce notations that are convenient as far as minimal transversality is concerned. Let  $S \subseteq V(\mathcal{H})$  and let  $v$  be a vertex in  $S$ . A *private edge of  $v$  with respect to  $S$*  is an edge  $E \in E(\mathcal{H})$  such that  $E \cap S = \{v\}$ . The set of private edges of  $v$  with respect to  $S$  are denoted by  $\text{priv}(S, v)$ . The following remark is folklore (see, e.g., the monograph [Ber84]).

**Remark 2.1.** *A transversal in a hypergraph is minimal if and only if each vertex it contains has a private edge.*

The *degeneracy* of a graph is a well-known parameter that has received considerable attention as a measure of sparsity in a graph [NOdM12], as well as a suitable parameter to devise tractable graph algorithms [EGM03, AG09, PRS12, WAU14]. It corresponds to the least integer  $d$  such that every subgraph has a vertex of degree at most  $d$ . Several generalizations to hypergraphs have been proposed in the literature [EGM03, KZ08, BNCOS10, DG16]. Among these generalizations, two of them naturally arise depending on the notion of subhypergraph we consider, and were previously considered in the context of hypergraph dualization [EGM03, CKMU19]. We review them here.

Let  $\mathcal{H}$  be a hypergraph and  $S$  a subset of vertices of  $\mathcal{H}$ . The hypergraph *induced* by  $S$  is the hypergraph

$$\mathcal{H}[S] := (S, \{E \in E(\mathcal{H}) : E \subseteq S\}).$$

The *trace* of  $\mathcal{H}$  on  $S$  is the hypergraph

$$\mathcal{H}_{|S} := (S, \{E \cap S : E \in E(\mathcal{H}), E \cap S \neq \emptyset\}).$$

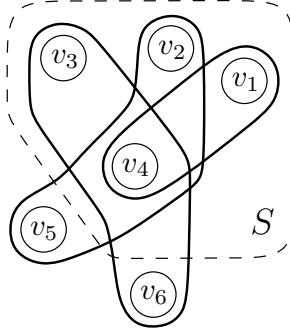


Figure 1: A hypergraph  $\mathcal{H}$  on 6 vertices and 3 edges. The hypergraph induced by  $S := \{v_1, v_2, v_3, v_4\}$  has a single edge  $\{v_1, v_4\}$ , while the trace of  $S$  on  $\mathcal{H}$  contains three edges  $\{v_1, v_4\}$ ,  $\{v_2, v_4\}$ , and  $\{v_3, v_4\}$ .

Note that, while several edges in  $\mathcal{H}$  may have the same intersection with  $S$ , that intersection is only counted once in  $\mathcal{H}|_S$ , *i.e.*, the trace defines a set and not a multiset. See Figure 1 for an example of these two notions of subhypergraph. The two following definitions generalize graph degeneracy (with the strong degeneracy being at most one more than the graph degeneracy), and differ on whether we consider the subhypergraph to be the hypergraph induced by the first  $i$  vertices or its trace on these vertices.

**Definition 1** (Weak degeneracy, [EGM03]). *The weak degeneracy of  $\mathcal{H}$  is the smallest integer  $d$  such that there exists an ordering  $v_1, \dots, v_n$  of its vertices satisfying for every  $1 \leq i \leq n$ :*

$$\deg_{\mathcal{H}[\{v_1, \dots, v_i\}]}(v_i) \leq d.$$

**Definition 2** (Strong degeneracy, [KZ08, CKMU19]). *The strong degeneracy of  $\mathcal{H}$  is the smallest integer  $d$  such that there exists an ordering  $v_1, \dots, v_n$  of its vertices satisfying for every  $1 \leq i \leq n$ :*

$$\deg_{\mathcal{H}|_{\{v_1, \dots, v_i\}}}(v_i) \leq d.$$

The next inequality follows from the fact that, for any subset  $S$  of  $V(\mathcal{H})$ , an edge  $E$  that belongs to  $\mathcal{H}[S]$  also belongs to  $\mathcal{H}|_S$  by definition.

**Remark 2.2.** *The weak degeneracy of a hypergraph is at most its strong degeneracy.*



Note that the strong degeneracy of a hypergraph can be arbitrarily larger than its weak degeneracy. To see this, consider the hypergraph  $\mathcal{H}$  formed by taking a clique on  $k$  vertices  $v_1, \dots, v_k$  (a graph), and adding one new vertex  $u$  in each edge, so that each hyperedge in  $\mathcal{H}$  has size 3. Let  $U$  be the set of vertices  $u$  added in the construction, and consider any ordering of  $V(\mathcal{H})$  such that the elements  $v_1, \dots, v_k$  appear before those in  $U$ . The weak degeneracy of  $\mathcal{H}$  is 1 since each vertex  $u$  has degree 1 and the hypergraph induced by  $v_1, \dots, v_k$  has no edge. On the other hand, the strong degeneracy of  $\mathcal{H}$  is at least  $k$  since the trace of  $\mathcal{H}$  on  $v_1, \dots, v_k$  is a clique whose vertices all have degree  $k$ .

Also, note that while there exist  $n$ -vertex hypergraphs with  $O(2^n)$  edges, any hypergraph that has (weak or strong) degeneracy  $d$  contains  $O(dn)$  edges. This explains why time bounds such as those expected in Questions 1.2, 1.3, and obtained in Theorem 1.4 and subsequent lemmas in this paper are still reasonable.

For both notions of degeneracy, an *elimination ordering* is an order  $v_1, \dots, v_n$  as in their definitions. We note that Question 1.2 is originally stated in the context of strong degeneracy in [CKMU19]. In this paper, we will show Theorem 1.4 to hold for weak degeneracy, which implies the same result in the context of strong degeneracy.

**Graph domination.** If a hypergraph is a graph, then its transversals are usually referred to as vertex covers in the literature. This is not to be confused with domination, that we define now. Let us consider some graph  $G$ . The *closed neighborhood* of a vertex  $v$  of  $G$  is the set  $N[v] := \{v\} \cup \{u : \{u, v\} \in E(G)\}$ . A *dominating set* in  $G$  is a subset  $D \subseteq V(G)$  intersecting every closed neighborhood in the graph. It is called *minimal* if it is minimal by inclusion, and the set of minimal dominating sets of  $G$  is denoted by  $D(G)$ . In the minimal dominating sets enumeration problem, denoted DOM-ENUM, the task is to generate  $D(G)$  given  $G$ .

As put in the introduction, both TRANS-ENUM and DOM-ENUM are polynomially equivalent, one direction being a direct consequence of the easy observation that  $D(G) = \text{Tr}(\mathcal{N}(G))$  for any graph  $G$ , where  $\mathcal{N}(G)$  denotes the *hypergraph of closed neighborhoods* of  $G$  whose vertex set is  $V(G)$  and whose edge set is  $\{N[v] : v \in V(G)\}$  (see, e.g., [Ber84]). The other direction was exhibited by Kanté *et al.* in [KLMN14] even when restricted to co-bipartite graphs. We note, however, that the construction given in [KLMN14] does not preserve the degeneracy. Hence, the existence of an FPT-delay algorithm for DOM-ENUM parameterized by the degeneracy (and more) would not directly imply one for TRANS-ENUM. This paper is devoted to giving a positive answer to Question 1.3, which thus may be regarded as an intermediate step toward answering Question 1.2, a point that is discussed in Section 6.

### 3 Ordered generation

Our algorithm is based on the *ordered generation* technique that was initially introduced in [EGM02] for monotone Boolean dualization, and later adapted to the context of graph domination in [BDHR19, BDH<sup>+</sup>20]. Our presentation of the algorithm is essentially based on these works.

Let  $\mathcal{H}$  be a hypergraph and  $v_1, \dots, v_n$  an ordering of its vertex set. For readability, given any  $i \in \{1, \dots, n\}$ , we set  $V_i := \{v_1, \dots, v_i\}$ ,

$$\mathcal{H}_i := \mathcal{H}[V_i],$$

and denote by  $\text{inc}_i(u)$ , for any  $u \in V(\mathcal{H}_i)$ , the incident edges of  $u$  in  $\mathcal{H}_i$ . By convention, we set  $V_0 := \emptyset$ ,  $\mathcal{H}_0 := (\emptyset, \emptyset)$ , and  $\text{Tr}(\mathcal{H}_0) := \{\emptyset\}$ . Further, for all  $0 \leq i \leq n$ ,  $S \subseteq V(\mathcal{H})$ , and  $v \in S$ , we denote by  $\text{priv}_i(S, v)$  the private edges of  $v$  with respect to  $S$  in  $\mathcal{H}_i$ . Let us now consider some integer  $i \in \{0, \dots, n-1\}$ . For  $T \subseteq \text{Tr}(\mathcal{H}_{i+1})$ , we call the *parent of  $T$  with respect to  $i+1$* , denoted by  $\text{parent}(T, i+1)$ , the set obtained by applying the following greedy procedure:

*While there exists a vertex  $v \in T$  such that  $\text{priv}_i(T, v) = \emptyset$ ,  
remove from  $T$  a vertex of smallest index with that property.*

Observe that  $\text{parent}(T, i+1)$  is uniquely defined by this routine. Given  $T^*$  a minimal transversal of  $\mathcal{H}_i$ , we will call the *children of  $T^*$  with respect to  $i$* , denoted by  $\text{children}(T^*, i)$ , the family of sets  $T \in \text{Tr}(\mathcal{H}_{i+1})$  such that  $\text{parent}(T, i+1) = T^*$ .

**Lemma 3.1.** *Let  $i \in \{0, \dots, n-1\}$ ,  $T \in \text{Tr}(\mathcal{H}_{i+1})$ . Then,  $\text{parent}(T, i+1) \in \text{Tr}(\mathcal{H}_i)$ .*

*Proof.* By definition, every edge of  $\mathcal{H}_i$  is an edge of  $\mathcal{H}_{i+1}$ . Hence,  $T$  is a transversal of  $\mathcal{H}_i$  as well. Now, as the greedy procedure above only removes vertices  $v$  that have no private edge in  $\mathcal{H}_i$ , the obtained set is a transversal of  $\mathcal{H}_i$  at each step, and ends up minimal by construction.  $\square$

**Lemma 3.2.** *Let  $i \in \{0, \dots, n-1\}$  and  $T^* \in \text{Tr}(\mathcal{H}_i)$ . Then, either:*

- $T^* \in \text{Tr}(\mathcal{H}_{i+1})$  and  $\text{parent}(T^*, i+1) = T^*$ ; or
- $T^* \cup \{v_{i+1}\} \in \text{Tr}(\mathcal{H}_{i+1})$  and  $\text{parent}(T^* \cup \{v_{i+1}\}, i+1) = T^*$ .

*Proof.* Note that as  $T^* \in \text{Tr}(\mathcal{H}_i)$ , we have  $\text{priv}_i(T^*, v) \neq \emptyset$  for every  $v \in T^*$ , and since every edge of  $\mathcal{H}_i$  is an edge of  $\mathcal{H}_{i+1}$ , we derive that  $\text{priv}_{i+1}(T^*, v) \neq \emptyset$  for every  $v \in T^*$ . Hence, if  $T^*$  is a transversal of  $\mathcal{H}_{i+1}$ , then we obtain  $T^* \in \text{Tr}(\mathcal{H}_{i+1})$ , and thus,  $\text{parent}(T^*, i+1) = T^*$  by definition, yielding the first item of the lemma.

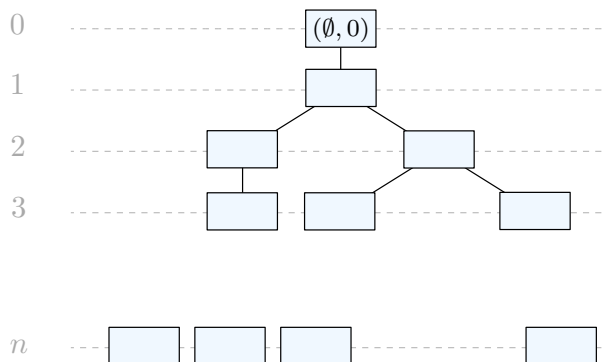


Figure 2: A representation of the solution tree defined by the parent relation.

Let us thus assume that  $T^*$  is not a transversal of  $\mathcal{H}_{i+1}$ . Then, there is an edge  $E$  in  $\mathcal{H}_{i+1}$  that is not hit by  $T^*$ . As  $T^*$  is a transversal of  $\mathcal{H}_i$ , that edge must contain the vertex  $v_{i+1}$ . We derive that  $T := T^* \cup \{v_{i+1}\}$  is a transversal of  $\mathcal{H}_{i+1}$ , which furthermore satisfies  $\text{priv}_{i+1}(T, v_{i+1}) \neq \emptyset$  as  $E$  is not hit by  $T^*$ . Let us show that  $\text{priv}_{i+1}(T, v) \neq \emptyset$  holds for any other vertex  $v \in T$ . Consider one such vertex  $v \neq v_{i+1}$  and  $E \in \text{priv}_i(T^*, v)$ . Such an edge exists as  $T^* \in \text{Tr}(\mathcal{H}_i)$ . If  $E \notin \text{priv}_{i+1}(T, v)$ , then we have that  $v_{i+1} \in E$ , which contradicts the fact that  $E$  belongs to  $\mathcal{H}_i$ . Hence,  $E \in \text{priv}_{i+1}(T, v)$ , and the lemma follows.  $\square$

Note that the parent relation defines a rooted tree, that we call the *solution tree*, on vertex set  $\{(T, i) : T \in \text{Tr}(\mathcal{H}_i), 0 \leq i \leq n\}$  with leaves  $\{(T, n) \mid T \in \text{Tr}(\mathcal{H})\}$  and root  $(\emptyset, 0)$ . Our generation algorithm will consist in a traversal of that tree starting from its root and such that we output every leaf. Each set  $T$  of an internal node  $(T, i)$  will be referred to as a *partial solution*. Lemma 3.1 ensures that each partial solution in the tree can be obtained from its parent. Lemma 3.2 ensures that each partial solution in the tree has at least one child, *i.e.*, that every branch of the tree leads to a different minimal transversal of  $\mathcal{H}$ . See Figure 2 for a representation of that tree. The overall complexity of the algorithm will depend on the delay needed to generate the children of an internal node, which is formalized in the next theorem.

**Theorem 3.3.** *Let  $f, s: \mathbb{N}^2 \rightarrow \mathbb{N}^*$  be two functions. Suppose that there is an algorithm that, given a hypergraph  $\mathcal{H}$  on vertices  $v_1, \dots, v_n$  and  $m$  edges, an integer  $i \in \{0, \dots, n-1\}$ , and  $T^* \in \text{Tr}(\mathcal{H}_i)$ , enumerates  $\text{children}(T^*, i)$  with delay  $f(n, m)$  and using  $s(n, m)$  space. Then, there is an algorithm that, given a hypergraph  $\mathcal{H}$  on  $n$  vertices and  $m$  edges, enumerates the set  $\text{Tr}(\mathcal{H})$  with delay  $O(n \cdot f(n, m))$  and total space  $O(n \cdot s(n, m))$ .*

*Proof.* We assume the existence of an algorithm **A** of the first type, and describe an algorithm **B** listing  $Tr(\mathcal{H})$  with the desired delay. The algorithm first chooses an ordering  $v_1, \dots, v_n$  of the vertex set of  $\mathcal{H}$  in  $O(n)$  time. Then, it runs a DFS of the solution tree defined by the parent relation above, starting from its root  $(\emptyset, 0)$  and only outputting sets  $T$  corresponding to visited leaves  $(T, n)$ . Recall that these sets are exactly the minimal transversals of  $\mathcal{H}$ . First, the correctness of **B** follows from Lemmas 3.1 and 3.2. Let us now consider the time and space complexity of such an algorithm. At first, the root  $(\emptyset, 0)$  is constructed in  $O(1)$  time and space. Then, for each internal node  $(T^*, i)$  of the tree, the set  $\text{children}(T^*, i)$  is generated with delay  $f(n, m)$  and space  $s(n, m)$  by **A**. We furthermore note that when visiting a node of the solution tree, we do not need to generate all its children. We can instead compute one child that has not been processed yet, and continue the DFS on that child. Thus, when we visit some node  $(T, i)$ , we only need to store the data of the  $i$  paused executions of **A** enumerating the children of the ancestors of  $(T, i)$  including the root  $(\emptyset, 0)$ , together with the data of the current instance of **A** listing  $\text{children}(T, i)$ . Hence, the maximum delay of **B** between two consecutive outputs is bounded by twice the height of the solution tree times the delay of **A**, and the same goes for the space complexity. Since the height of the tree is  $n$ , we get the desired bounds of  $O(n \cdot f(n, m))$  and  $O(n \cdot s(n, m))$  for the delay and space complexities of **B**, respectively.  $\square$

We note that only a multiplicative factor of  $f(n, m)$  is added to the input size in the delay of the algorithm given by Theorem 3.3. In particular, we get that the existence of an algorithm generating children with FPT-delay  $f(k) \cdot \text{poly}(n, m)$ , for some parameter  $k$  and computable function  $f$ , yields an FPT-delay algorithm for TRANS-ENUM.

## 4 Children generation parameterized by the weak degeneracy and dimension

In hypergraphs, we show that the children of an internal node of the solution tree, as defined in Section 3, may be enumerated with FPT-delay parameterized by the weak degeneracy and the dimension. By Theorem 3.3, this leads to an algorithm listing minimal transversals in hypergraphs with FPT-delay parameterized by the weak degeneracy and the dimension, *i.e.*, Theorem 1.4.

Let  $\mathcal{H}$  be a hypergraph of weak degeneracy  $d$  and dimension  $k$ , and let  $v_1, \dots, v_n$  be the elimination ordering of  $\mathcal{H}$  witnessing the degeneracy. Then, by definition, for

any  $i \in \{0, \dots, n-1\}$  and  $E \in \text{inc}_{i+1}(v_{i+1})$ , we have  $\text{inc}_{i+1}(v_{i+1}) \leq d$  and  $|E| \leq k$ . Let us fix  $i \in \{0, \dots, n-1\}$ . A subset  $R$  of  $\text{inc}_{i+1}(v_{i+1})$  is called a *trace* on  $\text{inc}_{i+1}(v_{i+1})$  if there exists a vertex  $w \in V_{i+1}$  whose set of incident edges in  $\text{inc}_{i+1}(v_{i+1})$  is precisely  $R$ . The next lemma is a direct consequence of  $\text{inc}_{i+1}(v_{i+1}) \leq d$ .

**Lemma 4.1.** *The number of distinct traces on  $\text{inc}_{i+1}(v_{i+1})$  is at most  $2^d$ .*

We define an equivalence relation  $\equiv_{i+1}$  on vertices in  $V_{i+1}$ , where  $u \equiv_{i+1} w$  if  $u$  and  $w$  intersect the same set of edges in  $\text{inc}_{i+1}(v_{i+1})$ .

**Lemma 4.2.** *Let  $T^* \in \text{Tr}(\mathcal{H}_i)$ . For  $X \subseteq V_{i+1} \setminus T^*$ , if  $T^* \cup X \in \text{Tr}(\mathcal{H}_{i+1})$ , then  $|X| \leq d$  and  $X$  contains at most one vertex per equivalence class of  $\equiv_{i+1}$ .*

*Proof.* First, note that no edge in  $\mathcal{H}_i$  may be the private edge of a vertex  $x \in X$  as all such edges are hit by  $T^*$ . Hence, if  $T^* \cup X \in \text{Tr}(\mathcal{H}_{i+1})$ , then every  $x \in X$  has a private edge in  $\text{inc}_{i+1}(v_{i+1})$ . Hence, no two such  $x$  belong to a same equivalence class of  $\equiv_{i+1}$ , and we conclude that  $|X| \leq d$  by recalling that  $\text{inc}_{i+1}(v_{i+1}) \leq d$ .  $\square$

Note that as  $|E| \leq k$  for any edge  $E$  in  $\text{inc}_{i+1}(v_{i+1})$ , we furthermore get that each equivalence class of  $\equiv_{i+1}$  contains at most  $k$  elements. We derive the following.

**Lemma 4.3.** *The set  $\text{children}(T^*, i)$  can be generated in  $O(2^d + (k+1)^{2^d} \cdot n^{O(1)})$  time.*

*Proof.* First, we compute the equivalence classes of  $\equiv_{i+1}$  in  $2^d \cdot n^{O(1)}$  time by Lemma 4.1 (recall that  $m = O(dn)$ ). Then, we try all possible ways of selecting  $X$ , that is, selecting at most one vertex per equivalence class of  $\equiv_{i+1}$ , in total time

$$(k+1)^{2^d}$$

as each of these classes has size at most  $k$ . For every such set  $X$ , we check whether the obtained set  $T := T^* \cup X$  is a minimal transversal of  $\mathcal{H}_{i+1}$  satisfying  $T^* = \text{parent}(T, i+1)$  in  $n^{O(1)}$  time by definition. Clearly, all these steps can be done within the claimed total time. By Lemma 4.2, this is an exhaustive search for the children of node  $(T^*, i)$  in the solution tree.  $\square$

We note that the dependence in  $d$  on  $k$  in the time bound specified in Lemma 4.3 may be improved by only selecting elements in at most  $d$  equivalence classes of  $\equiv_{i+1}$ , at the cost of an increased dependence on  $d$  in the first term of the time bound. Specifically, this other approach yields an algorithm for listing the children of an internal node  $(T^*, i)$  in time

$$\sum_{i=0}^d \binom{2^d}{d} \cdot k^d \cdot n^{O(1)}.$$

Additionally, it should be noted that simpler arguments may also be used as long as an FPT-delay and polynomial-space algorithm for TRANS-ENUM parameterized by the weak degeneracy and dimension is concerned. Indeed, since the set of elements belonging to edges incident to  $v_{i+1}$  is of size at most  $kd$ , the set  $X$  can be chosen among  $2^{kd}$  subsets of vertices. Note that we do not need to store all these sets  $X$  since they can be generated following a lexicographic order on the vertices, and thus, this approach only needs polynomial space in  $n$ . The analysis and time presented in Lemma 4.3 are chosen to match the bounds given in Questions 1.2 and 1.3. Theorem 1.4 then follows by Theorem 3.3 and Lemma 4.3, by noting that  $(k+1)^{2d} = O(k^{f(d)})$  for some computable function  $f$ .

## 5 Consequences for minimal domination

Let  $G$  be a graph of degeneracy  $d$  and maximum degree  $\Delta$ . We show that, as a corollary of Theorem 1.4, we obtain an FPT-delay algorithm listing the minimal dominating sets of  $G$  parameterized by  $d$  and  $\Delta$ , answering Question 1.3.

In the following, let  $v_1, \dots, v_n$  be the elimination ordering of  $G$  witnessing the degeneracy. Note that  $v_1, \dots, v_n$  witnesses the weak degeneracy of  $\mathcal{N}(G)$  as well. Indeed, for any  $i \in \{1, \dots, n\}$ , the number of hyperedges in  $\text{inc}_i(v_i)$  is bounded by the number of neighbors of  $v_i$  in  $\{v_1, \dots, v_{i-1}\}$ . We furthermore have the following relation between the dimension of  $\mathcal{N}(G)$  and  $\Delta$ .

**Lemma 5.1.** *The dimension of  $\mathcal{N}(G)$  is at most the maximum degree of  $G$  plus one.*

*Proof.* Let  $E$  be a hyperedge in  $\mathcal{N}(G)$ . Then,  $E = N[u]$  for some vertex  $u$  in  $G$ , and hence,  $|E|$  is at most the degree of  $u$  plus one, which in turn is bounded by  $\Delta+1$ .  $\square$

As a corollary of Theorem 1.4 and Lemma 5.1, we conclude that the aforementioned algorithm exists, yielding a positive answer to Question 1.3.

## 6 Discussions

We exhibited an FPT-delay algorithm for TRANS-ENUM parameterized by the weak degeneracy and dimension, yielding an FPT-delay algorithm for DOM-ENUM parameterized by the degeneracy and maximum degree, answering a question in [CKMU19]. This is, to the best of our knowledge, the first FPT-delay algorithm for minimal dominating sets enumeration, and it is tempting to ask whether the problem admits an FPT-delay algorithm for other natural parameters. However, we note that, for well-studied parameters such as clique number in graphs [BDH<sup>+</sup>20] or poset dimension

in comparability graphs [BDMN20], even the existence of an XP-delay algorithm for DOM-ENUM remains open.

A natural strengthening of Question 1.3 would be to get rid of the maximum degree on the required time bound. Formally, we may ask the following.

**Question 6.1.** *Can the minimal dominating sets of an  $n$ -vertex  $d$ -degenerate graph be listed with delay  $f(d) \cdot n^{O(1)}$  for some computable function  $f$ ?*

The same strengthening can be formulated for Question 1.2 as well. However, it is valuable to note that the techniques presented here seem to fail in the context of hypergraphs, as an FPT-delay algorithm listing children parameterized by the degeneracy (only) yields an FPT-delay algorithm for MULTICOLORED INDEPENDENT SET parameterized by the number of colors, a notorious  $W[1]$ -hard problem [CFK<sup>+</sup>15]. Recall that, in MULTICOLORED INDEPENDENT SET, the vertices of a graph  $G$  are colored with colors  $1, \dots, k$ , and we have to find an independent set, *i.e.*, a set of pairwise non-adjacent vertices, containing exactly one vertex from each color. We construct an instance of children generation as follows. From  $G$ , for each color  $1 \leq \ell \leq k$ , we create a hyperedge  $E_\ell$  consisting of the vertices of color  $\ell$  together with some extra vertex  $v$  that each of  $E_1, \dots, E_k$  will contain. For each edge  $xy \in E(G)$ , we add a vertex  $u_{xy}$  and two hyperedges  $A_{xy} := \{x, u_{xy}\}$  and  $B_{xy} := \{y, u_{xy}\}$ . This completes the description of the hypergraph we denote by  $\mathcal{H}$ . It has weak degeneracy  $k \geq 2$  as witnessed by the elimination ordering obtained by first removing  $v$ , then removing the  $u_{xy}$ 's, and then removing all of the remaining vertices in an arbitrary order. By setting  $n$  to be the number of vertices of  $\mathcal{H}$ ,  $i := n - 1$ , and  $v_{i+1} := v$ , we get that, for all  $1 \leq \ell \leq k$  and  $xy \in E(G)$ ,  $E_\ell$  belongs to  $\text{inc}_{i+1}(v_{i+1})$  and  $A_{xy}, B_{xy} \subseteq V_i$ . Let  $T^* := \{u_{xy} : xy \in E(G)\}$ . Clearly, by construction,  $T^* \in \text{Tr}(\mathcal{H}_i)$  and  $A_{xy}, B_{xy}$  are the only private edges of each  $u_{xy} \in T^*$ . Let us now consider the generation of the children of  $T^*$ . For  $X \subseteq V_{i+1} \setminus T^*$  to be such that  $T^* \cup X \in \text{Tr}(\mathcal{H}_{i+1})$ , it must be that  $X$  hits all of  $E_1, \dots, E_k$ . Furthermore, each such hyperedge must be hit once, as every vertex it contains has the same trace on  $\text{inc}_{i+1}(v_{i+1})$ . Furthermore,  $X$  should not contain the two endpoints of any  $xy \in E(G)$ , as otherwise the corresponding vertex  $u_{xy} \in T^*$  would lose its private neighbors. Hence, among all possible such candidate sets  $X$ , one consists of the singleton  $\{v_{i+1}\}$ , and the others are multicolored independent sets of  $G$ , completing the reduction.

Finally, another natural strengthening of Theorem 1.4 would be to relax the degeneracy, and require to be FPT-delay in the dimension only. We note however that even the existence of an XP-delay algorithm in that context seems open [KBEG07].



**Acknowledgements.** The second author is thankful to Mamadou Kanté for bringing Questions 1.2 and 1.3 to his attention, and to Arnaud Mary for pointing out the flaw in the arXiv preprint [Wil23].

## References

- [AG09] Noga Alon and Shai Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. *Algorithmica*, 54(4):544–556, 2009.
- [BDH<sup>+</sup>20] Marthe Bonamy, Oscar Defrain, Marc Heinrich, Michał Pilipczuk, and Jean-Florent Raymond. Enumerating minimal dominating sets in  $K_t$ -free graphs and variants. *ACM Transactions on Algorithms*, 16(3):39:1–39:23, 2020.
- [BDHR19] Marthe Bonamy, Oscar Defrain, Marc Heinrich, and Jean-Florent Raymond. Enumerating minimal dominating sets in triangle-free graphs. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:12. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.
- [BDMN20] Marthe Bonamy, Oscar Defrain, Piotr Micek, and Lhouari Nourine. Enumerating minimal dominating sets in the (in)comparability graphs of bounded dimension posets. *arXiv preprint arXiv:2004.07214*, 2020.
- [Ber84] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45 of *North-Holland Mathematical Library*. Elsevier, 1984.
- [BM16] Endre Boros and Kazuhisa Makino. Generating maximal irredundant and minimal redundant subfamilies of a given hypergraph. In *WEPA 2016 & Boolean Seminar 2017*, 2016.
- [BNCOS10] Amotz Bar-Noy, Panagiotis Cheilaris, Svetlana Olonetsky, and Shakhar Smorodinsky. Online conflict-free colouring for hypergraphs. *Combinatorics, Probability and Computing*, 19(4):493–516, 2010.
- [CFK<sup>+</sup>15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.



- [CKM<sup>+</sup>19] Nadia Creignou, Raïda Ktari, Arne Meier, Julian-Steffen Müller, Frédéric Olive, and Heribert Vollmer. Parameterised enumeration for modification problems. *Algorithms*, 12(9):189, 2019.
- [CKMU19] Alessio Conte, Mamadou Moustapha Kanté, Andrea Marino, and Takeaki Uno. Maximal irredundant set enumeration in bounded-degeneracy and bounded-degree hypergraphs. In *Combinatorial Algorithms: 30th International Workshop, IWOCA 2019, Pisa, Italy, July 23–25, 2019, Proceedings*, pages 148–159. Springer, 2019.
- [CMM<sup>+</sup>17] Nadia Creignou, Arne Meier, Julian-Steffen Müller, Johannes Schmidt, and Heribert Vollmer. Paradigms for parameterized enumeration. *Theory of Computing Systems*, 60:737–758, 2017.
- [Dam06] Peter Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351(3):337–350, 2006.
- [DG16] Kunal Dutta and Arijit Ghosh. On subgraphs of bounded degeneracy in hypergraphs. In *Graph-Theoretic Concepts in Computer Science: 42nd International Workshop, WG 2016, Istanbul, Turkey, June 22–24, 2016, Revised Selected Papers*, pages 295–306. Springer, 2016.
- [EG95] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.
- [EGM02] Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New results on monotone dualization and generating hypergraph transversals. In *Proceedings of the thirty-fourth annual ACM Symposium on Theory of Computing*, pages 14–22, 2002.
- [EGM03] Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM Journal on Computing*, 32(2):514–537, 2003.
- [EMG08] Thomas Eiter, Kazuhisa Makino, and Georg Gottlob. Computational aspects of monotone dualization: A brief survey. *Discrete Applied Mathematics*, 156(11):2035–2049, 2008.

- [Fer02] Henning Fernau. On parameterized enumeration. In *Computing and Combinatorics: 8th Annual International Conference, COCOON 2002 Singapore, August 15–17, 2002 Proceedings 8*, pages 564–573. Springer, 2002.
- [FGS19] Henning Fernau, Petr A. Golovach, and Marie-France Sagot. Algorithmic enumeration: Output-sensitive, input-sensitive, parameterized, approximative (dagstuhl seminar 18421). In *Dagstuhl Reports*, volume 8. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
- [FK96] Michael L. Fredman and Leonid Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3):618–628, 1996.
- [GHK<sup>+</sup>16] Petr A. Golovach, Pinar Heggernes, Mamadou M. Kanté, Dieter Kratsch, and Yngve Villanger. Enumerating minimal dominating sets in chordal bipartite graphs. *Discrete Applied Mathematics*, 199:30–36, 2016.
- [GHK<sup>+</sup>18] Petr A. Golovach, Pinar Heggernes, Mamadou M. Kanté, Dieter Kratsch, Sigve H. Sæther, and Yngve Villanger. Output-polynomial enumeration on graphs of bounded (local) linear MIM-width. *Algorithmica*, 80(2):714–741, 2018.
- [GKKL22] Petr A. Golovach, Christian Komusiewicz, Dieter Kratsch, and Van B. Le. Refined notions of parameterized enumeration kernels with applications to matching cut enumeration. *Journal of Computer and System Sciences*, 123:76–102, 2022.
- [GMKT97] Dimitrios Gunopulos, Heikki Mannila, Roni Khardon, and Hannu Toivonen. Data mining, hypergraph transversals, and machine learning. In *PODS*, pages 209–216. ACM, 1997.
- [JYP88] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.
- [KBEG07] Leonid Khachiyan, Endre Boros, Khaled Elbassioni, and Vladimir Gurvich. On the dualization of hypergraphs with bounded edge-intersections and other related classes of hypergraphs. *Theoretical computer science*, 382(2):139–150, 2007.

- [KLM<sup>+</sup>15] Mamadou M. Kanté, Vincent Limouzy, Arnaud Mary, Lhouari Nourine, and Takeaki Uno. A polynomial delay algorithm for enumerating minimal dominating sets in chordal graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 138–153. Springer, 2015.
- [KLMN14] Mamadou M. Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. On the enumeration of minimal dominating sets and related notions. *SIAM Journal on Discrete Mathematics*, 28(4):1916–1929, 2014.
- [KPS93] Dimitris Kavvadias, Christos H. Papadimitriou, and Martha Sideri. On horn envelopes and hypergraph transversals. In *International Symposium on Algorithms and Computation*, pages 399–405. Springer, 1993.
- [KZ08] Alexandr V. Kostochka and Xuding Zhu. Adapted list coloring of graphs and hypergraphs. *SIAM Journal on Discrete Mathematics*, 22(1):398–408, 2008.
- [Mei18] Arne Meier. Enumeration in incremental FPT-time. *arXiv preprint arXiv:1804.07799*, 2018.
- [Mei20] Arne Meier. Incremental FPT delay. *Algorithms*, 13(5):122, 2020.
- [NOdM12] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.
- [NP12] Lhouari Nourine and Jean-Marc Petit. Extending set-based dualization: Application to pattern mining. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 630–635. IOS Press, 2012.
- [PRS12] Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms (TALG)*, 9(1):1–23, 2012.
- [Str19] Yann Strozecki. Enumeration complexity. *Bulletin of EATCS*, 1(129), 2019.
- [WAU14] Kunihiro Wasa, Hiroki Arimura, and Takeaki Uno. Efficient enumeration of induced subtrees in a k-degenerate graph. In *Algorithms and*

*Computation: 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings*, pages 94–102. Springer, 2014.

- [Wil23] Marcel Wild. Enumerating all minimal hitting sets in polynomial total time. *arXiv preprint arXiv:2303.07708*, 2023.