



**HAL**  
open science

## On the enumeration of signatures of XOR-CNF's

Nadia Creignou, Oscar Defrain, Frédéric Olive, Simon Vilmin

► **To cite this version:**

Nadia Creignou, Oscar Defrain, Frédéric Olive, Simon Vilmin. On the enumeration of signatures of XOR-CNF's. The Algorithms and Data Structures Symposium, Aug 2025, Toronto, Canada. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, <10.4230/LIPIcs.WADS.2025.19>. <hal-04502055>

**HAL Id: hal-04502055**

**<https://hal.science/hal-04502055v1>**

Submitted on 13 Mar 2024

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.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# On the enumeration of signatures of XOR-CNF's

Nadia Creignou, Oscar Defrain, Frédéric Olive, and Simon Vilmin

Aix-Marseille Université, CNRS, LIS, Marseille, France.

February 28, 2024

## Abstract

Given a CNF formula  $\varphi$  with clauses  $C_1, \dots, C_m$  over a set of variables  $V$ , a truth assignment  $\mathbf{a} : V \rightarrow \{0, 1\}$  generates a binary sequence  $\sigma_\varphi(\mathbf{a}) = (C_1(\mathbf{a}), \dots, C_m(\mathbf{a}))$ , called a signature of  $\varphi$ , where  $C_i(\mathbf{a}) = 1$  if clause  $C_i$  evaluates to 1 under assignment  $\mathbf{a}$ , and  $C_i(\mathbf{a}) = 0$  otherwise. Signatures and their associated generation problems have given rise to new yet promising research questions in algorithmic enumeration. In a recent paper, Bérczi et al. interestingly proved that generating signatures of a CNF is tractable despite the fact that verifying a solution is hard. They also showed the hardness of finding maximal signatures of an arbitrary CNF due to the intractability of satisfiability in general. Their contribution leaves open the problem of efficiently generating maximal signatures for tractable classes of CNFs, i.e., those for which satisfiability can be solved in polynomial time. Stepping into that direction, we completely characterize the complexity of generating all, minimal, and maximal signatures for XOR-CNFs.

**Keywords:** algorithmic enumeration, XOR-CNF, signatures, maximal bipartite subgraphs enumeration, extension, proximity search.

## 1 Introduction

Propositional formulas are ubiquitous in computer science. The complexity of the satisfiability problem has been extensively studied from the point of view of several algorithmic tasks such as decidability, counting, or enumeration to mention but a few. Given a CNF formula  $\varphi$  with clauses  $C_1, \dots, C_m$  over a set of variables  $V = \{v_1, \dots, v_n\}$ , a truth assignment  $\mathbf{a} : V \rightarrow \{0, 1\}$  leads to a binary sequence  $\sigma_\varphi(\mathbf{a})$ , called a *signature* of  $\varphi$ , defined by  $\sigma_\varphi(\mathbf{a}) = (C_1(\mathbf{a}), \dots, C_m(\mathbf{a}))$  where  $C_i(\mathbf{a}) = 1$  if clause  $C_i$  evaluates to 1 under assignment  $\mathbf{a}$ , and  $C_i(\mathbf{a}) = 0$  otherwise. Deciding the satisfiability of  $\varphi$  boils down to deciding whether the all-one sequence is a signature of  $\varphi$ .

In this paper, we investigate the problems of listing all, minimal, and maximal signatures of a CNF, where minimal and maximal are meant bitwise. The task of finding all signatures of a CNF originates from well-design pattern trees and has first been posed during the Dagstuhl seminar on enumeration in data management [BKPS19]. However, enumerating the signatures of a given CNF—may it be all, minimal, or maximal ones—will generally produce an exponential number of solutions. Therefore, input sensitive polynomial time complexity is not a suitable yardstick of efficiency when analyzing algorithms performance. Instead, *output-sensitive complexity* estimates the running time of an enumeration algorithm using both input and output size. More precisely, an enumeration algorithm runs in *total-polynomial time* (or *output-polynomial time*) if its execution time is bounded by a polynomial in the combined size of the input and the output. Still, the regularity of enumeration algorithms is more relevant than their total running time. For this reason, *polynomial delay* and *incremental-polynomial time* are customarily regarded as better notions of tractability for enumeration complexity. On the one hand, polynomial delay means that the delay between consecutive outputs is polynomial in the input size. Incremental-polynomial time, on the other hand, means that the time to produce the  $i^{\text{th}}$  solution is bounded by a polynomial in the input size plus  $i$  (see e.g., [JYP88, Str19]).

Generating all the signatures of a CNF turns out to be of great fundamental interest, for it is an example of enumeration problem which can be solved in total-polynomial time even though solutions cannot be recognized in polynomial time.<sup>1</sup> In a recent contribution [BBČ+21], Bérczi et al. address this problem. They give an incremental-polynomial time algorithm for listing all the signatures of a CNF, and improve their result to polynomial delay for tractable CNF's. On the other hand, they show that generating maximal signatures is hard, while minimal signatures can be listed with polynomial delay, regardless of the CNF under consideration. Their positive result relies on the equivalence with maximal independent sets enumeration in graphs, known to be tractable [TIAS77]. As of their hardness result, it relies on the fact that the existence of at least two solutions would imply the non-satisfiability of the CNF at hand. Thus, the difficulty of the enumeration heavily relies on the intractability of the satisfiability of CNF's in general. This observation naturally leads to the following open question, where by tractable CNF's we mean families of CNF's for which the satisfiability of a formula  $\varphi$  in the family (as well as its sub-formulas) can be decided in polynomial time.

**Question 1.** *Can the maximal signatures of tractable CNF's be enumerated in total-polynomial time?*

Among tractable formulas, Horn and 2-CNF's are two natural candidates for approaching Question 1. These two cases however proved to be very challenging, and the question of their tractability was explicitly stated as an open problem in the 2022's edition of the Workshop on Enumeration Problems and Applications (WEPA'22).

---

<sup>1</sup>These problems thus lie outside of the class EnumP which is commonly studied in algorithmic enumeration; see e.g. [Str19].

Another natural step toward answering Question 1 is to consider XOR-CNF formulas, being conjunctions of clauses using the “exclusive-or” connective instead of the usual “or” connective. On top of being tractable for satisfiability, XOR-CNF’s appear in many complexity classifications and enjoy polynomial time algorithms for a broad variety of problems including counting and enumerating their models (see e.g., [CH96, CH97, COS11]). Note that these formulas do not fall in the framework of CNF’s though. Indeed, rewriting a XOR-CNF as a CNF may result in a blowup on the number of clauses and signatures, as discussed in Section 2. Hence, we remark that the positive results from [BBČ+21] do not directly apply to XOR-CNF. Yet, we show that signatures enumeration remains tractable using flashlight search as it is done in [BBČ+21]. Namely, we prove the following.

**Theorem 1.1.** *The set of signatures of a XOR-CNF formula can be enumerated with polynomial delay and polynomial space.*

Concerning the enumeration of minimal signatures, we show that the problem is in fact equivalent to the case of listing maximal signatures.

**Proposition 1.2.** *There is polynomial-delay and polynomial space algorithm listing all minimal signatures of a XOR-CNF formula if and only if there is one listing all maximal signatures of a XOR-CNF formula.*

This leave us with the study of maximal signatures enumeration for XOR-CNF’s. Quite surprisingly, we prove this problem to relate well with other problems from graph theory and matroid theory. Specifically, we show the case of 2-XOR-CNF to generalize maximal bipartite subgraphs enumeration, a problem which was recently shown to admit a polynomial-delay algorithm [CGM+22]. As of the general case of XOR-CNF’s, it can be seen as listing all maximal satisfiable subsystems of a linear system of equations, which is related to the enumeration of circuits passing through a given element in a matroid [BEGK03]; see also [KBE+05, KBB+09]. Relying on these results, we derive that maximal signatures enumeration for XOR-CNF’s is tractable. Namely, we obtain the following.

**Theorem 1.3.** *There is an incremental-polynomial time algorithm generating the maximal (or minimal) signatures of a XOR-CNF.*

Concerning the case of 2-XOR-CNF, we show it to be even more tractable, that is, we show that it can be solved with polynomial delay using proximity search.

**Theorem 1.4.** *There is a polynomial-delay algorithm generating the maximal (or minimal) signatures of a 2-XOR-CNF.*

To prove Theorem 1.4, we represent the input 2-XOR-CNF as an edge-bicolored multigraph where each color codes the parity of the XOR clause. Then, the enumeration amounts to list maximal bipartite subgraphs with additional constraints on colored edges.

A caveat of Theorems 1.3 and 1.4 is that a queue of solutions has to be maintained in the generation. More precisely, in [BEGK03] the queue is needed to generate a new solution as the algorithm uses the saturation technique where new solutions are derived from obtained ones. In [CGM<sup>+</sup>22], the queue has the other purpose of preventing repetitions. In both cases, getting rid of a potentially exponential space use is a natural enhancement one may seek. It should however be noted that improving the algorithm from [BEGK03] to get polynomial space is open for two decades now. As of improving Theorem 1.4 to use polynomial space, it would imply such a result for maximal bipartite subgraphs enumeration, another open question [CGM<sup>+</sup>22]. Toward such directions, we prove that the folklore technique of flashlight search used to obtain Theorem 1.1 may probably not be of great use as deciding whether a subsequence of clauses can be extended into a signature is NP-complete.

**Organization of the paper.** In the next section we introduce the notions that will be used in this paper, basic properties, and prove Theorem 1.1. The equivalence between maximal and minimal signatures enumeration for XOR-CNF is detailed in Section 3. Theorems 1.3 and 1.4 are proved in Section 4. Future directions and the limitations of flashlight search toward an improvement of Theorem 1.4 are finally discussed in Section 5.

## 2 Preliminaries

All the objects considered in this paper are finite. If  $V$  is a set,  $2^V$  is its powerset. Let  $m \in \mathbb{N}$  and let  $\sigma \in \{0, 1\}^m$  be a binary sequence. We write  $\sigma[i]$  to denote the value of the  $i^{\text{th}}$  element of  $\sigma$ . It will be furthermore convenient to note  $\mathbf{1}(\sigma)$  the set of indices  $I \subseteq \{1, \dots, m\}$  whose corresponding values in  $\sigma$  is 1;  $\mathbf{0}(\sigma)$  is defined analogously. The *complementary* of  $\sigma$  is the binary sequence  $\bar{\sigma}$  obtained from  $\sigma$  by flipping all the bits, i.e.,  $\bar{\sigma}[i] = 1 + \sigma[i] \pmod{2}$  for  $1 \leq i \leq m$ . If  $\sigma, \tau \in \{0, 1\}^m$ , we write  $\sigma \leq \tau$  if  $\sigma[i] \leq \tau[i]$  for all each  $1 \leq i \leq m$ . In other words,  $\leq$  corresponds to the bitwise order.

**Enumeration complexity.** An enumeration algorithm aims at enumerating a set of solutions of some problem, one after the other, with no repetition. We already have defined relevant time complexities in the introduction. However, space is also an important consideration in the analysis of enumeration algorithms. We assume that the solutions we produce are not stored in memory but rather flashed and discarded. Thus, when measuring space we only consider the space needed by the algorithm in order to conduct the enumeration, in terms of the input size. We refer to [JYP88, Str19] for more details on the complexity of enumeration algorithms.

**Graphs.** Given an undirected graph  $G$ , we write  $V(G)$  its set of vertices and  $E(G)$  its set of edges. For convenience, we will note  $uv$  instead of  $\{u, v\}$  for edges of  $G$ . A graph

$G$  is bipartite if there exists a bipartition  $V_1, V_2$  of  $V(G)$  such that each edge of  $E(G)$  has one endpoint in  $V_1$  and the other in  $V_2$ . Equivalently, a graph  $G$  is bipartite if and only if it does not contain any odd cycles. In this paper we will consider *multigraphs* as well, meaning graphs with potentially several edges sharing two same endpoints, i.e., with  $E(G) = \{e_1, \dots, e_m\}$  a multiset. In this case and to avoid any ambiguity we will refer to the edges of  $G$  by their label rather than their endpoints.

**XOR-formulas.** We assume basic familiarity with propositional logic. Let  $V = \{x_1, \dots, x_n\}$  be a set of Boolean variables. A *literal* is a variable  $x_i$  (positive literal) or its negation  $\neg x_i$  (negative literal). A *clause*  $C$  is a disjunction of literals. The size  $|C|$  of  $C$  is its number of literals. We say that  $C$  is a  $k$ -clause if  $|C| \leq k$ . A formula in conjunctive normal form (or a CNF for short)  $\varphi$  is a conjunction of clauses, i.e.,  $\varphi = C_1 \wedge \dots \wedge C_m$  where  $C_i$ , for  $1 \leq i \leq m$ . It will be convenient to denote by  $V(\varphi)$  the variables of  $\varphi$ . The (total) *size* of  $\varphi$  is defined as  $\sum_{1 \leq i \leq m} |C_i|$ . An assignment of the variables in  $V$  is a mapping  $\mathbf{a} : V \rightarrow \{0, 1\}$ . Given an assignment  $\mathbf{a}$ , we write  $C_i(\mathbf{a}) = 1$  (resp.  $C_i(\mathbf{a}) = 0$ ) if  $C_i$  evaluates to 1 (resp. 0) under assignment  $\mathbf{a}$ . The notations  $\varphi(\mathbf{a}) = 1$  and  $\varphi(\mathbf{a}) = 0$  are defined analogously. The formula  $\varphi$  is *satisfiable* if there exists an assignment  $\mathbf{a}$  of  $V(\varphi)$  such that  $\varphi(\mathbf{a}) = 1$ ; it is unsatisfiable otherwise.

A *XOR-clause* is a clause in which the usual connective “or” is replaced by the “exclusive-or” connective. It is well-known that any XOR-clause can be represented as a linear equation  $x_1 + \dots + x_k = \varepsilon$ ,  $\varepsilon \in \{0, 1\}$  over the two-elements field  $\mathbf{F}_2$ . In particular, the 2-XOR-clause  $x_1 + x_2 = 0$  (resp.  $x_1 + x_2 = 1$ ) is satisfied if and only if  $x_1 = x_2$  (resp.  $x_1 \neq x_2$ ). A XOR-clause is *odd* (resp. *even*) if  $\varepsilon = 1$  (resp. 0). Given a XOR-clause  $C = (x_1 + \dots + x_k = \varepsilon)$ , with  $\varepsilon \in \{0, 1\}$ , we put  $\bar{C} = (x_1 + \dots + x_k = 1 - \varepsilon)$  and call  $\bar{C}$  the *negation* of  $C$ . In other words, the bar operator *changes the parity* of  $C$ . A XOR-CNF  $\varphi$  is a conjunction of XOR-clauses. Equivalently,  $\varphi$  can be seen as a system of linear equations over  $\mathbf{F}_2$ . For a XOR-CNF  $\varphi = C_1 \wedge \dots \wedge C_m$ , we denote by  $\bar{\varphi}$  the XOR-CNF  $\bar{C}_1 \wedge \dots \wedge \bar{C}_m$  and call it the *inverse* of  $\varphi$ . We recall that the satisfiability of a XOR-CNF can be tested in polynomial time; see e.g., [CH11, Theorem 2.18].

**Signatures.** The next notations and terminology are borrowed from [BBČ+21]. Let  $\varphi = C_1 \wedge \dots \wedge C_m$  be a CNF. We refer to the introduction for the definition of a signature, and shall say that an assignment  $\mathbf{a}$  of  $\varphi$  *produces* the signature  $\sigma_\varphi(\mathbf{a}) = (C_1(\mathbf{a}), \dots, C_m(\mathbf{a}))$ . We will further drop the subscript  $\varphi$  from this notation when it is clear from the context. Note that in particular,  $\varphi$  is satisfiable if and only if  $(1, \dots, 1)$  is a signature of  $\varphi$ . A signature  $\sigma$  of  $\varphi$  is *minimal* if it is minimal with respect to  $\leq$  among all signatures of  $\varphi$ . Similarly, we call *maximal* a signature which is maximal for  $\leq$  among signatures of  $\varphi$ .

In this paper we are interested in the following problems.

Signatures enumeration for XOR-CNF's.

**Input:** A XOR-CNF  $\varphi$ .

**Output:** The set of signatures of  $\varphi$ .

Maximal signatures enumeration for XOR-CNF's.

**Input:** A XOR-CNF  $\varphi$ .

**Output:** The set of maximal signatures of  $\varphi$ .

Minimal signatures enumeration for XOR-CNF's.

**Input:** A XOR-CNF  $\varphi$ .

**Output:** The set of minimal signatures of  $\varphi$ .

Observe that in the above problems, and by the previous discussions, we may indifferently consider  $\varphi$  as a set of XOR clauses or as a set of binary equations over  $\mathbf{F}_2$ . A subset of simultaneously satisfiable clauses on one side corresponds to a feasible subset of equations on the other side. In the rest of the paper, we will assume that  $\varphi$  is in this latter form. In particular, when  $\varphi$  is seen as a system of linear equations over  $\mathbf{F}_2$ , there is a one-to-one correspondence between maximal signatures of  $\varphi$  and maximal feasible subsets of equations.

Another remark is that any XOR-CNF may be rewritten as a CNF, i.e., the  $\oplus$  operator may be rewritten using  $\wedge$  and  $\vee$ . For example, the XOR-clause  $x_1 \neq x_2$  can be rewritten as  $(x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2)$ . We however note that we do not have a bijection between the sets of signatures of the two formulas. Typically, generalizing the above example to disjoint clauses will produce a XOR-CNF with one minimal signature while the equivalent CNF has exponentially many such signatures. Though this does not rule out the existence of a reduction between (maximal) signatures of XOR-CNF's to (maximal) signatures of CNF's it should be noted that the status of maximal signatures enumeration is still open for 2-CNF's, a point that is discussed in Section 5. As of the case of all signatures, it admits a simple and direct proof using flashlight search, as explained in Section 3.

We finally argue that duplicate clauses in  $\varphi$  may be ignored as far as the enumeration of signatures is concerned. Indeed, having one of these clauses to 1 implies that all other copies are to 1, and vice versa. Hence, we will assume without loss of generality that all formulas are without duplicated clauses in the rest of the paper, i.e., that they are defined as pairwise distinct XOR-clauses.

### 3 Signatures of XOR-CNF's and their properties

We prove that testing whether a binary sequence is a signature (resp. minimal signature, maximal signature) of a XOR-CNF can be done in polynomial time in the size of the CNF at hand, and use it to show that listing all signatures of a XOR-CNF is tractable. For the rest of the section, let us fix some XOR-CNF  $\varphi = C_1 \wedge \cdots \wedge C_m$  on variable set  $V = \{x_1, \dots, x_n\}$ . A preliminary step is the following.

**Proposition 3.1.** *A sequence  $\sigma \in \{0, 1\}^m$  is a signature of  $\varphi$  if and only if  $\bar{\sigma}$  is a signature of  $\bar{\varphi}$ . In particular,  $\sigma$  is a minimal (resp. maximal) signature of  $\varphi$  if and only if  $\bar{\sigma}$  is a maximal (resp. minimal) signature of  $\bar{\varphi}$ .*

*Proof.* Let  $\mathbf{a}$  be an assignment producing the signature  $\sigma$  of  $\varphi$ . By definition we have  $C_i(\mathbf{a}) = 1$  and  $C_j(\mathbf{a}) = 0$  for each  $i, j \in \{1, \dots, m\}$  such that  $\sigma[i] = 1$  and  $\sigma[j] = 0$ . This can be equivalently written as  $\bar{C}_i(\mathbf{a}) = 0$  if  $\bar{\sigma}[i] = 0$  and  $\bar{C}_i(\mathbf{a}) = 1$  when  $\bar{\sigma}[i] = 1$ . This is in turn equivalent to  $\mathbf{a}$  producing the signature  $\bar{\sigma}$  of  $\bar{\varphi}$ . The second part of the statement follows from the fact that for every sequences  $\sigma, \tau \in \{0, 1\}^m$  we have  $\sigma \leq \tau$  if and only if  $\bar{\sigma} \leq \bar{\tau}$ .  $\square$

From Proposition 3.1 we already derive Proposition 1.2.  
Given two disjoint subsets  $A, B \subseteq \{1, \dots, m\}$ , we define

$$\varphi(A, B) := \left( \bigwedge_{i \in A} C_i \right) \wedge \left( \bigwedge_{j \in B} \bar{C}_j \right)$$

to be the formula obtained from  $\varphi$  by taking the clauses with index in  $A$  and the negation of clauses with index in  $B$ . We furthermore set this formula to be defined on variable set  $V(\varphi)$  even though some variables in  $V(\varphi)$  may not appear in any of the clauses of  $\varphi(A, B)$ —we want this to ensure that assignments of  $\varphi(A, B)$  are well-defined for  $\varphi$ . Remark that the size of  $\varphi(A, B)$  is at most that of  $\varphi$ , and that it is defined on precisely  $m$  clauses whenever  $A, B$  is a bipartition of  $\{1, \dots, m\}$ . In the next proposition, we show that testing whether a binary sequence is a signature of a XOR-CNF can be reduced to satisfiability testing.

**Proposition 3.2.** *A sequence  $\sigma \in \{0, 1\}^m$  is a signature of  $\varphi$  if and only if the formula  $\varphi(\mathbf{1}(\sigma), \mathbf{0}(\sigma))$  is satisfiable.*

*Proof.* We start with the only if part. Assume that  $\sigma$  is a signature of  $\varphi$ . Then there is an assignment  $\mathbf{a}$  that produces  $\sigma$ . By definition  $\mathbf{a}$  satisfies the clauses  $C_i$  such that  $\sigma[i] = 1$ , and it does not satisfy the clauses  $C_j$  such that  $\sigma[j] = 0$ . This latter case is equivalent to  $\mathbf{a}$  satisfying the clauses  $\bar{C}_j$ . Hence  $\mathbf{a}$  is an assignment which satisfies the formula  $\varphi(\mathbf{1}(\sigma), \mathbf{0}(\sigma))$  as desired.

We move to the if part. Assume that  $\varphi(\mathbf{1}(\sigma), \mathbf{0}(\sigma))$  has a satisfying assignment  $\mathbf{a}$  defined over  $V$ . Now since  $\mathbf{a}$  satisfies the clauses  $C_i$  of  $\varphi$  such that  $\sigma[i] = 1$ , and not the clauses  $C_j$  of  $\varphi$  such that  $\sigma[j] = 0$ , we conclude that  $\sigma$  is indeed a signature of  $\varphi$  as required.  $\square$

**Proposition 3.3.** *Deciding whether  $\sigma$  is a minimal (resp. maximal) signature of  $\varphi$  can be done in polynomial time in the size of  $\varphi$ .*

*Proof.* To check that  $\sigma$  is a signature, we rely on Proposition 3.2: we build the XOR-CNF  $\varphi(\mathbf{1}(\sigma), \mathbf{0}(\sigma))$  and test its satisfiability, all of which in polynomial time.

We prove the if part. Suppose that  $\psi(j)$  is satisfiable for some  $j \in \mathbf{0}(\sigma)$ , and consider an arbitrary assignment  $\mathbf{a}$  over  $V$  satisfying  $\psi(j)$ . Let  $\tau$  be the signature of  $\varphi$  produced by assignment  $\mathbf{a}$ . Since  $\mathbf{a}$  satisfies  $\psi(j)$ , every 1 in  $\sigma$  is a 1 in  $\tau$ , so that  $\sigma \leq \tau$ . Moreover,  $\sigma[j] < \tau[j]$  holds, and  $\sigma < \tau$  thus follows. Hence  $\sigma$  is not a maximal signature concluding the if implication. As for the only if part, the existence of a signature  $\tau > \sigma$  implies by definition that  $\psi(j)$  is satisfiable for some  $j \in \mathbf{0}(\sigma)$  such that  $\tau[j] > \sigma[j]$ .

Henceforth, deciding whether  $\sigma$  is a maximal signature of  $\varphi$  can be done by  $m$  calls to the satisfiability of a XOR-CNF whose size is no greater than  $\varphi$ . We conclude that it can be done in polynomial time. For minimality, we conclude using Proposition 3.1.  $\square$

To conclude this section, we argue that enumerating all the signatures of a XOR-CNF is tractable using the above properties and flashlight search. We recall that the main idea of this technique is to conduct binary search over the space of solutions. In order for flashlight search to run with polynomial delay and polynomial space, it is sufficient to solve the so-called “extension problem” in polynomial time: given two disjoint subsets  $A, B$  of the groundset, decide whether there exists a solution including  $A$  and avoiding  $B$ . We refer to e.g. [BEG04, MS19] for a more detailed description of this folklore technique.

In the case of XOR-CNF signatures,  $A$  will consist of indices of clauses to be satisfied in the signature—the 1’s in the sequence—while  $B$  will consist of indices of clauses not to be satisfied—the 0’s in the sequence. Then, it follows from the discussion above that  $\varphi$  admits a signature  $\sigma$  satisfying  $\sigma[i] = 1$  and  $\sigma[j] = 0$  for all  $i \in A$  and  $j \in B$  if and only if  $\varphi(A, B)$  is satisfiable. Indeed, a satisfying assignment  $\mathbf{a}$  of  $\varphi(A, B)$  satisfies the clauses of  $\varphi$  with index in  $A$ , and not those with index in  $B$ . Thus the signature of  $\varphi$  produced by  $\mathbf{a}$  meets the requirements. As for the converse direction, it follows directly from the definition of a signature. We conclude by Proposition 3.3 that the extension problem can be solved in polynomial time for XOR-CNF signatures, hence proving Theorem 1.1 that we recall here.

**Theorem 1.1.** *The set of signatures of a XOR-CNF formula can be enumerated with polynomial delay and polynomial space.*

We point that Theorem 1.1 is not a direct consequence of [BBČ<sup>+</sup>21, Theorem 1]. Indeed, in the later theorem the authors consider families of tractable CNF’s which do not contain XOR-CNF’s in general. The arguments, however, follow the same line.

## 4 Maximal signatures of XOR-CNF’s

We investigate the complexity of enumerating the maximal signatures of a given XOR-CNF. More precisely, we prove the general case to be solvable in incremental-polynomial time by a result of Boros et al. [BEGK03], and the 2-XOR-CNF case to be solvable with polynomial delay using proximity search.

In Section 2 we have mentioned that the clauses of a given XOR-CNF  $\varphi$  may be seen as linear equations  $x_1 + \dots + x_k = \varepsilon$ ,  $\varepsilon \in \{0, 1\}$  over the two-elements field  $\mathbf{F}_2$ . Hence  $\varphi$  can be seen as a system of linear equations whose maximal feasible subsystems correspond to its maximal signatures. In [BEGK03], the authors present an incremental polynomial-time algorithm for enumerating all circuits of a matroid containing an element, and more generally, one for enumerating all maximal subsets not spanning an element. Using Farkas' lemma (more precisely the Fredholm alternative) it can be derived an algorithm which, given an infeasible system of linear equations, enumerates all its maximal feasible (or minimal infeasible) subsystems within the same time; see also [KBE<sup>+</sup>05, KBB<sup>+</sup>09] on the use of this characterization of infeasible systems. This immediately yields Theorem 1.3 that we restate here.

**Theorem 1.3.** *There is an incremental-polynomial time algorithm generating the maximal (or minimal) signatures of a XOR-CNF.*

We now focus on 2-XOR-CNF's. Recall from Section 2 that we can consider these formulas as a set of 2-clauses of the form  $x = y$  or  $x \neq y$  for positive literals  $x, y$  only, together with a set of 1-clauses ( $z$ ) with  $z$  a positive or negative literal. Furthermore we may assume that no clause is repeated in the formula, as far as signatures enumeration is concerned. We shall call *graphic* a 2-XOR-CNF with clauses of the first type only—that is, without 1-clauses—and will assume for now on that our XOR-CNF's are graphic. Later, we will show how to deal with 1-clauses with small modifications on the instance.

We start by defining an underlying multigraph structure of graphic 2-XOR-CNF that will be convenient for applying the framework of proximity search. Given such a XOR-CNF  $\varphi$  we define  $G(\varphi)$  (or simply  $G$  when it is clear from the context) as the edge-bicolored multigraph defined on vertex set  $V(\varphi)$  and where there is a blue edge  $xy$  whenever  $x \neq y$  is a clause of  $\varphi$ , and a red edge  $xy$  whenever  $x = y$  is a clause of  $\varphi$ . A same pair of variables  $x, y$  can produce both a red and a blue edge: this corresponds to having clauses  $x = y$  and  $x \neq y$  in  $\varphi$  which typically makes it unsatisfiable. However as no clause is repeated in  $\varphi$ , no two edges of a same color share the same endpoints. In other words, our multigraph has edge multiplicity at most two, and two edges between a same pair of endpoints means that one is blue and the other is red; see Figure 1 for an illustration of such multigraphs. In the following, we will label edges of  $G$  as  $e_1, \dots, e_m$  so that the edge  $e_j$  corresponds to the clause  $C_j$  in  $\varphi$ . We note  $R(G)$  the set of red edges of  $G$ , and  $B(G)$  the set of its blue edges. By definition, we have  $R(G) \cap B(G) = \emptyset$  and  $R(G) \cup B(G) = E(G)$ . A naive but crucial property we get from that partition is that the set of red edges characterizes the set of blue edge, and vice versa. By subgraph  $H$  of an edge-bicolored multigraph  $G$  we mean the graph on same vertex set  $V(H) = V(G)$  and  $E(H) \subseteq E(G)$ , that is, we consider subgraphs as sets of edges. Finally, given two disjoint subsets  $X, Y \subseteq V(\varphi)$ , by  $\delta(X, Y)$  we mean the set of edges having an endpoint in  $X$  and the other in  $Y$ .

Note that an arbitrary (simple, uncolored) graph  $H$  is bipartite whenever it admits a bipartition  $(X, Y)$  of its vertex set with  $E(H) = \delta(X, Y)$ . In the next lemma,

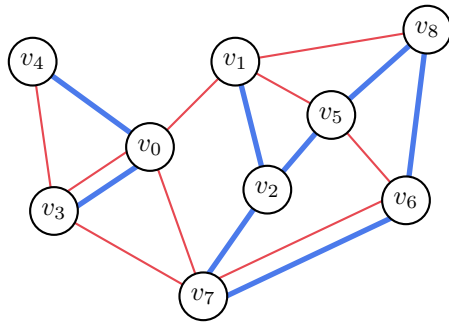


Figure 1: An edge-bicolored multigraph  $G$  on nine vertices, with red edges denoted by thin red lines and blue edges denoted by bold blue lines.

we relate the satisfiability of a graphic 2-XOR-CNF to a property of its underlying edge-bicolored multigraph which generalizes bipartiteness. We stress the fact that this property may only be stated on blue edges, which is still reasonable as blue and red edges partition the edges of the graph.

**Lemma 4.1.** *Let  $\varphi$  be a graphic 2-XOR-CNF. Then  $\varphi$  is satisfiable if and only if there is a bipartition  $(X, Y)$  of the vertex set of  $G(\varphi)$  such that  $B(G) = \delta(X, Y)$ . Moreover, every such bipartition corresponds to a model  $\mathbf{a}$  of  $\varphi$  with  $X = \mathbf{a}^{-1}(1)$  and  $Y = \mathbf{a}^{-1}(0)$ .*

*Proof.* We prove the first implication. If  $\varphi$  is satisfiable then it has a satisfying assignment  $\mathbf{a}$  defining a bipartition  $(X, Y)$  of  $V(\varphi)$  where  $X = \mathbf{a}^{-1}(1)$  and  $Y = \mathbf{a}^{-1}(0)$ . Clearly  $B(G) \subseteq \delta(X, Y)$  as otherwise we have two variables  $x, y$  with  $x \neq y$  yet  $\mathbf{a}(x) = \mathbf{a}(y)$ . On the converse, if there is a red edge in  $\delta(X, Y)$  we deduce the existence of two variables  $x, y$  with  $x = y$  yet  $\mathbf{a}(x) \neq \mathbf{a}(y)$ . Hence,  $\delta(X, Y) \subseteq E(G) \setminus R(G) = B(G)$  and  $B(G) = \delta(X, Y)$  follows. This concludes the first implication.

We prove the other implication. Let  $(X, Y)$  be a bipartition of  $V(G)$  such that  $B(G) = \delta(X, Y)$ . Consider the assignment  $\mathbf{a}$  defined by  $\mathbf{a}(x) = 1$  and  $\mathbf{a}(y) = 0$  for all  $x \in X$  and  $y \in Y$ . By construction to each clause of  $\varphi$  of the form  $x \neq y$  corresponds a blue edge lying in  $\delta(X, Y)$  and hence such that  $\mathbf{a}(x) \neq \mathbf{a}(y)$ . As of the clauses of the form  $x = y$  they correspond to red edges not in  $\delta(X, Y)$  and hence such that  $\mathbf{a}(x) = \mathbf{a}(y)$ . All these clauses are satisfied by  $\mathbf{a}$  and we conclude that  $\varphi$  is indeed satisfiable.  $\square$

**Lemma 4.2.** *Let  $\varphi$  be a graphic 2-XOR-CNF. Then the maximal signatures of  $\varphi$  are exactly the maximal subgraphs  $H$  of  $G(\varphi)$  admitting a bipartition  $(X, Y)$  of their vertex set such that  $\delta(X, Y) = B(H)$ .*

*Proof.* Let  $\sigma$  be a maximal signature of  $\varphi$  and fix  $H := G(\psi)$  for  $\psi := \varphi(\mathbf{1}(\sigma), \emptyset)$ —recall that  $\psi$  is defined on the full variable set  $V(\varphi)$  while some of its variables may not appear in any clause. Since  $\psi$  is satisfiable by definition of a signature, and using Lemma 4.1, we derive that  $B(H) = \delta(X, Y)$  for a bipartition  $(X, Y)$  of  $V(G)$  corresponding to a model  $\mathbf{a}$  of  $\psi$ . Suppose that  $H$  is not maximal with this property. Then there exist

an edge  $e_j \in E(G) \setminus E(H)$ ,  $1 \leq j \leq m$  and a bipartition  $(X', Y')$  with  $H' := H + e_j$  satisfying  $B(H') = \delta(X', Y')$ . Then by Lemma 4.1 we derive that  $\psi := \varphi(\mathbf{1}(\sigma) \cup \{j\}, \emptyset)$  is satisfiable. Let  $\mathbf{a}$  be an assignment satisfying  $\psi$ . Since  $j$  belongs to  $\mathbf{0}(\sigma)$ , we obtain that  $\mathbf{a}$  produces a signature  $\tau$  with  $\tau > \sigma$ , a contradiction.

Let  $H$  be a maximal subgraph of  $G$  admitting a bipartition  $(X, Y)$  of its vertex set such that  $B(G) = \delta(X, Y)$ . By Lemma 4.1 the formula  $\psi := \varphi(\{j : e_j \in E(H)\}, \emptyset)$  corresponding to the edges of  $H$  is satisfiable. Let  $\mathbf{a}$  be a satisfying assignment of  $\psi$  and consider the signature  $\sigma$  it produces for  $\varphi$ . We argue that it is a maximal signature of  $\varphi$ . Suppose that this is not the case and let  $\tau$  be a signature of  $\varphi$  such that  $\tau > \sigma$ . Then  $\psi' := \varphi(\mathbf{1}(\sigma) \cup \{j\}, \emptyset)$  is satisfiable for some  $1 \leq j \leq m$  with  $\sigma[j] < \tau[j]$ . Hence by Lemma 4.1, the graph  $H' := G(\psi')$  satisfies  $B(H') = \delta(X', Y')$ . Moreover  $E(H) \subset E(H')$  which contradicts the fact that  $H$  is chosen maximal. We conclude that  $\sigma$  is a maximal signature.  $\square$

In the following, given an edge-bicolored multigraph, we call *red-blue bipartite* any of its subgraph with the property described in Lemma 4.2. In Figure 2, we give an example of a maximal red-blue bipartite subgraph  $H$  of the multigraph  $G$  of Figure 1. Note that red-blue bipartite subgraphs do not contain both a red edge and a blue edge between a same pair of vertices, i.e., they define edge-bicolored graphs.

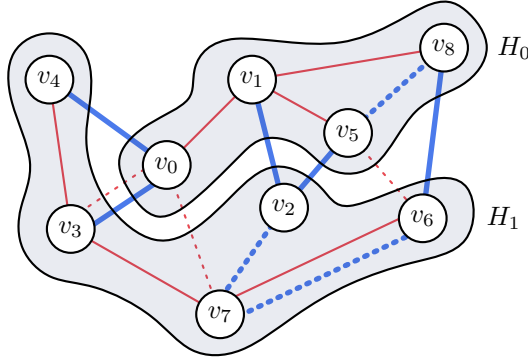


Figure 2: A maximal (connected) red-blue bipartite subgraph  $H$  of the multigraph  $G$  defined in Figure 1. The bipartition  $(H_0, H_1)$  of  $H$  is highlighted. The edges of  $G$  not in  $H$  are dotted.

**Theorem 4.3.** *There is a polynomial-delay and polynomial-space algorithm enumerating the maximal signatures of a graphic 2-XOR-CNF  $\varphi$  if and only if there is one listing the maximal red-blue bipartite subgraphs of its underlying multigraph  $G(\varphi)$ .*

*Proof.* This is a consequence of Lemma 4.2 observing that constructing either the formula or the edge-bicolored multigraph can be done in polynomial time in the size of the structure at hand. As there is a bijection between the two solutions sets this reduction preserves polynomial delay and polynomial space.  $\square$

A consequence of Lemma 4.2 is that the maximal signatures of a 2-XOR-CNF only consisting of disequalities  $x \neq y$  is equivalent to the enumeration of maximal (edge) bipartite subgraphs of a given graph. Indeed in that case,  $R(G)$  is empty—hence it is a graph—and we only require  $E(G) = \delta(X, Y)$ . We derive the following using a recent result by Conte and Uno [CGM<sup>+</sup>22] on maximal bipartite subgraphs enumeration using proximity search.

**Corollary 4.4.** *There is a polynomial-delay algorithm listing the maximal signatures of a 2-XOR-CNF consisting of disequalities  $x \neq y$  only.*

We end the section showing that the result from [CGM<sup>+</sup>22] can be extended into our context using proximity search, and arguing how to capture clauses of size one.

In the following, we say that an edge-bicolored multigraph  $G$  is *connected* if its underlying graph (where colors are ignored and multi-edges are considered as a single edge) is connected. Note that we can assume without loss of generality that  $G(\varphi)$  is connected, as far as signatures enumeration is concerned. Indeed, if it was not the case we could enumerate the signature of each subformula of  $\varphi$  corresponding to connected components of  $G(\varphi)$  and combine the results by doing their Cartesian product. Hence, we may further assume that  $G(\varphi)$  is connected, and shall call *connected* a formula such that its underlying multigraph is. We first show that if  $G$  is connected then its maximal red-blue bipartite subgraphs are as well, yielding a canonical bipartition.

**Lemma 4.5.** *Let  $G$  be a connected edge-bicolored multigraph. Then any of its maximal red-blue bipartite subgraph is connected.*

*Proof.* Let  $H$  be an arbitrary maximal red-blue bipartite subgraph of  $G$  and suppose toward a contradiction that it is not connected. Since  $G$  is connected we can find a shortest path  $P$  connecting two distinct and connected components  $C_1, C_2$  of  $H$ . For convenience, we consider two such components minimizing the length of  $P$  so that adding  $P$  only connects  $C_1, C_2$ . We obtain a red-blue bipartite supergraph of  $H$  by the following procedure: we start from the endpoint of  $P$  in  $C_1$  and for each of the edges  $e_1, \dots, e_k$  of  $P$  in order, we either keep its other endpoint in the same part if  $e_i$  is red, or put its endpoint in the other part if  $e_i$  is blue. When reaching  $e_k$ , we may need to swap the original bipartition of  $C_2$ . No contradiction can be found during this process and the obtained graph is indeed a supergraph of  $H$ . This contradicts the fact that  $H$  is maximal and we conclude that  $H$  is connected as desired.  $\square$

**Lemma 4.6.** *If  $G$  is a connected edge-bicolored bipartite graph then it admits a unique bipartition  $(X, Y)$  of its vertex set with  $B(G) = \delta(X, Y)$  and  $X$  containing the vertex of smallest label in  $G$ .*

*Proof.* We prove the statement using induction on the number of vertices of  $G$ . If  $G$  has 0, 1, or 2 vertices, the result is clear. Now, assume that the statement holds for every graph with at most  $k$  vertices. Let  $G$  be a connected edge-bicolored bipartite graph with vertices  $V(G) = \{v_1, \dots, v_{k+1}\}$ . We consider the edge-colored bipartite induced

subgraph  $H := G[\{v_1, \dots, v_k\}]$  with  $v_{k+1}$  chosen so that  $H$  is connected. Note that this can be done by considering a BFS of  $G$  launched at its vertex  $v_1$  of smallest label, and choosing  $v_{k+1}$  as the vertex obtained last in the traversal. By inductive hypothesis, there exists a unique bipartition  $(X', Y')$  of  $V(H)$  such that  $\delta(X', Y') = B(H)$  and  $v_1 \in X'$ . Since  $G$  is bipartite, there exists at least one bipartition  $(X, Y)$  of  $V(G)$  such that  $\delta(X, Y) = B(G)$ . Moreover, every such bipartition must comply with  $(X', Y')$ , that is, without loss of generality,  $X' \subseteq X$  and  $Y' \subseteq Y$ . Henceforth,  $G$  has at most two possible valid bipartitions:  $(X', Y' \cup \{v_{k+1}\})$  and  $(X' \cup \{v_{k+1}\}, Y')$ . Now,  $G$  is connected, so there exists a vertex  $v_i$  in  $V(G) \setminus \{v_{k+1}\}$ , say in  $X'$ , such that  $v_i v_{k+1}$  is an edge of  $G$ . We have two disjoint cases:

- $v_i v_{k+1}$  is red in which case  $(X' \cup \{v_{k+1}\}, Y')$  is the unique correct bipartition; and
- $v_i v_{k+1}$  is blue in which case  $(X', Y' \cup \{v_{k+1}\})$  is the unique correct bipartition.

We deduce using induction that there exists exactly one bipartition  $(X, Y)$  of  $G$  satisfying  $\delta(X, Y) = B(G)$  with  $X$  containing the vertex of smallest label in  $G$ .  $\square$

Proximity search has been introduced in [CU19] as a special case of solution graph traversal in which the proof that the solution graph is strongly connected relies on a notion of proximity between solutions that is more involved than their intersection. More precisely, the proximity between two solutions  $S, S^*$  is defined as the length of the longest prefix of  $S^*$  (according to a carefully chosen ordering of its elements) that is subset of  $S$ , and need not to be symmetric. We refer to [CGM<sup>+</sup>22] for more details on the technique. In [CGM<sup>+</sup>22], the authors extract the necessary conditions that make an enumeration problem “proximity searchable” and thus to be solvable with polynomial delay. We restate these conditions here in the context of XOR-CNF for self-containment and better readability.

In the reformulation below, let  $G$  be an edge-bicolored multigraph and  $\mathcal{H}(G)$  denote the set of maximal (connected) red-blue bipartite subgraphs of  $G$  we shall call *maximal solutions*. We further call *solutions* the connected red-blue bipartite subgraphs of  $G$ .

**Theorem 4.7** (Reformulation of [CGM<sup>+</sup>22, Definition 4.2 and Theorem 4.3]). *For arbitrary  $H, H^* \in \mathcal{H}(G)$ , let  $\mu(H^*)$  denote an ordering of the edges in  $H^*$ , and  $H \tilde{\cap} H^*$  denote the elements in the longest prefix of  $\mu(H^*)$  that is a subset of  $H$ . Then, there is a polynomial-delay algorithm enumerating  $\mathcal{H}(G)$  whenever:*

1. *For  $H \in \mathcal{H}(G)$  any prefix of  $\mu(H)$  is a solution;*
2. *There is a polynomial-time computable function COMP, which given a solution  $H'$  produces  $H = \text{COMP}(H')$  a maximal solution such that  $H \supseteq H'$ ;*
3. *Given  $H \in \mathcal{H}(G)$  and  $e \in E(G) \setminus E(H)$  there is a family  $\mathcal{K} \subseteq 2^{E(H)}$  of sets called *removables* such that:*
  - *the family  $\mathcal{K}$  can be computed in polynomial time;*

- for any  $K \in \mathcal{K}$  the set  $H \setminus K \cup \{e\}$  is a solution; and
- for any  $H^* \in \mathcal{H}(G)$  and minimal element  $e$  in  $\mu(H^*)$  that is not in  $H$ , there exists  $K \in \mathcal{K}$  with  $(H \tilde{\cap} H^*) \cap K = \emptyset$ .

The rest of the section is dedicated to proving that the conditions of Theorem 4.7 are fulfilled for well-chosen  $\mu$  and set  $\mathcal{K}$ . The fact that COMP can be computed in polynomial time follows from the fact that the red-blue bipartiteness is a hereditary property which can be tested in polynomial time, hence that we can obtain a maximal solution by a greedy procedure.

We start with the definition of  $\mu$ . Given a maximal red-blue bipartite graph  $H$  of  $G$  we denote by  $(H_0, H_1)$  the unique bipartition given by Lemma 4.6 where  $H_0$  is the side containing the vertex  $v_0$  of smallest index in  $G$ . Consider a BFS ordering of the vertices of  $G$  starting at  $v_0$ :  $v_0$  is first in the order and at each step, given the last vertex in the ordering, we add its neighborhood in ascending order of their label. Clearly, each prefix of such an ordering defines a connected induced subgraph. We define  $\mu(H)$  as the increasing ordering of the edges of  $H$  with respect to their endpoint occurring later in the BFS ordering launched at  $v_0$ , and break the tie by increasing order of their earlier endpoint in the BFS; see Figure 3 for an example of such an edge-ordering. This choice of an edge-ordering of a maximal solution yields the following property which will be crucial in the rest of the proof.

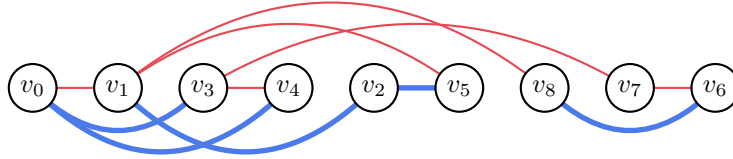


Figure 3: A BFS on the maximal red-blue bipartite subgraph  $H$  of  $G$  (Figure 2) will give the following order on the vertices:  $v_0, v_1, v_3, v_4, v_2, v_5, v_8, v_7, v_6$ . The order of the edges of  $H$  it produces is  $\mu(H) = v_0v_1, v_0v_3, v_0v_4, v_3v_4, v_1v_2, v_1v_5, v_2v_5, v_1v_8, v_3v_7, v_8v_6, v_7v_6$ .

**Observation 4.8.** *If  $H \in \mathcal{H}(G)$  then any prefix of  $\mu(H)$  is a connected red-blue bipartite subgraph of  $G$  and the first edge in this ordering is incident to  $v_0$ .*

Let us now consider arbitrary  $H, H^* \in \mathcal{H}(G)$  and put  $H' := H \tilde{\cap} H^*$  as defined in Theorem 4.7. Note that if  $H'$  is empty then the trivial removable  $K = E(G)$  satisfies the above conditions. Hence we may assume for convenience (in the rest of the analysis) that  $|H \tilde{\cap} H^*| \neq 0$ , while it will appear clear later that this assumption is in fact not needed. By Observation 4.8 since the first edge in  $\mu(H^*)$  belongs to both  $H', H$  and  $H^*$  we derive that  $v_0$  belongs to all three graphs. Again by Observation 4.8,  $H'$  is a connected red-blue bipartite subgraph of  $G$ . Thus by Lemma 4.6 it admits a unique bipartition  $(H'_0, H'_1)$  with  $v_0 \in H'_0$ . Now since  $H$  and  $H^*$  contain  $H'$  they must agree with that bipartition, meaning that  $H'_0 \subseteq H_0 \cap H_0^*$  and  $H'_1 \subseteq H_1 \cap H_1^*$ , which is a crucial point. In the following, for each edge  $e = ab$  in  $E(G) \setminus E(H)$ , we define

$K_1 := \{av : av \in E(G)\}$  and  $K_2 := \{bv : bv \in E(G)\}$  to be the two removables of  $(H, e)$  and argue that  $\mathcal{K} := \{K_1, K_2\}$  meets the requirements of the theorem.

Consider the first edge  $e = ab$  in  $\mu(H^*)$  that is not an edge of  $H$ . We distinguish two symmetric cases on whether  $e$  is red or blue, and only detail the blue case here. This situation is depicted in Figure 4.

We assume without loss of generality that the endpoint  $a$  belongs to  $H_0^*$  and that the other endpoint  $b$  lies in  $H_1^*$ . Recall that  $e \notin H$ . Hence since  $e$  is blue, either the two endpoints  $a, b$  belong to  $H_0$ , or they both belong to  $H_1$ . Two symmetric cases arise. Let us assume  $\{a, b\} \subseteq H_0$ . Note that in  $H^*$  we have no blue edge between  $b$  and  $H_1'$ , and no red edge between  $b$  and  $H_0'$ . In  $H$  this is the opposite: we have no blue edge between  $b$  and  $H_0'$ , and no red edge between  $b$  and  $H_1'$ . Now as  $E(H') \subseteq E(H^*) \cap E(H)$  we conclude that  $b$  is not incident to blue or red edges in  $H'$ , that is,  $E(H') \cap K_2 = \emptyset$ . This concludes the case and the other situation of  $\{a, b\} \subseteq H_1$  yields the symmetric situation of  $E(H') \cap K_1 = \emptyset$  by the same arguments. The other situation of  $e$  being red follows the exact same line of arguments, swapping red for blue.

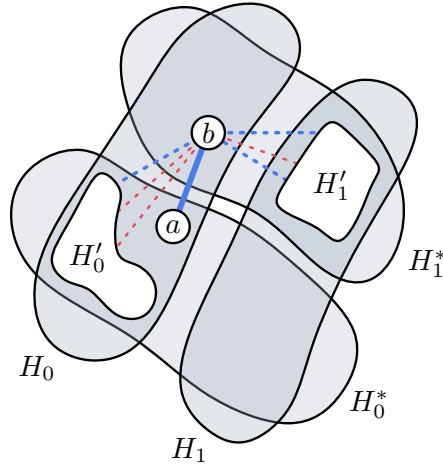


Figure 4: The situation occurring when considering two solutions  $H, H^*$  in Theorem 4.7 with  $H' = H \tilde{\cap} H^*$ :  $H_0' \subseteq H_0 \cap H_0^*$ ,  $H_1' \subseteq H_1 \cap H_1^*$  and the blue edge  $ab$  of  $H^*$  satisfies  $E(H') \cap K_2 = \emptyset$  where  $K_2 := \{bv : bv \in E(G)\}$ .

We finish the proof observing that the graphic conditions of 2-XOR-CNF's can be relaxed. As a preliminary step, note that 1-clauses containing variables that do not appear in 2-clauses can be removed from the instance as they will be set to true in any maximal signature. Thus we may assume that  $\varphi$  does not contain such isolated 1-clauses. We code every other 1-clauses  $(x)$  and  $(\bar{y})$ —which can also be seen as clauses  $x = 1$  and  $y = 0$ —by a blue edge  $xu$  and a red edge  $yu$  in the edge-bicolored multigraph  $G$  we have defined above, for some special vertex  $u$  that will be connecting all such 1-clauses. Note that  $G$  stays connected by the above assumption that  $\varphi$  has no isolated 1-clause. Then we set  $u$  to be the vertex of smallest label in  $G$ , followed by  $v_0$ . Then by the definition of  $\mu(H)$  the vertex  $u$  will be placed in  $H_0$  which can be seen as forcing the

gadget to false, satisfying the clause  $y = 0$  if the red edge  $uy$  is selected, and satisfying the clause  $x = 1$  if the blue edge  $ux$  is selected. We conclude with Theorem 1.4 that we restate here as a consequence of Theorem 4.7 and the above discussion.

**Theorem 1.4.** *There is a polynomial-delay algorithm generating the maximal (or minimal) signatures of a 2-XOR-CNF.*

## 5 Discussions

In this work we have showed that enumerating all (resp. minimal, maximal) signatures of a XOR-CNF formula is tractable, namely, Theorems 1.1, 1.3 and 1.4. Observe that the algorithm of Theorem 1.3 runs in incremental-polynomial time while the other two run with polynomial delay. Hence it is natural to ask whether Theorem 1.3 can be improved to polynomial delay.

**Question 2.** *Can the maximal signatures of a XOR-CNF be generated with polynomial delay?*

The algorithm by Boros et al. [BEGK03] underlying Theorem 1.3 relies on the enumeration of the circuits of a matroid, for which the same question seems open since two decades. Hence, answering Question 2 would either require new techniques, or would need to answer the open question from [BEGK03].

Furthermore, except for Theorems 1.1 and 1.3 that run with polynomial space, the algorithm of Theorem 1.4 requires a space that is potentially exponential as solutions must be stored into a queue. A natural question is the following.

**Question 3.** *Can the maximal signatures of a 2-XOR-CNF be generated with polynomial delay and polynomial space?*

We however stress the fact that a positive answer to Question 3 would improve the algorithm by Conte et al. for maximal bipartite subgraphs enumeration for which the questions of achieving polynomial delay and space is open [CGM<sup>+</sup>22].

Toward this direction, we now argue that flashlight search may not be adapted in order to give a positive answer to Question 3, as the extension problem is hard in that case. Moreover, this result even holds when restricted to 2-XOR-CNF's consisting of disequalities, that is, in the context of maximal bipartite subgraph enumeration, which may be of independent interest.

**Theorem 5.1.** *The problem of deciding, given a graph  $G$  and two disjoint edge sets  $A, B \subseteq E(G)$ , whether there exists a maximal bipartite (edge) subgraph  $H$  of  $G$  such that  $A \subseteq E(H)$  and  $B \cap E(H) = \emptyset$  is NP-complete.*

*Proof.* We first argue that the problem belongs to NP. Note that bipartiteness is hereditary, meaning that every subgraph of a bipartite graph is bipartite. Moreover since

bipartiteness can be decided in polynomial time, we can check maximality of  $H$  by trying to add every edge in  $E(G) \setminus E(H)$  and checking whether the resulting graph is bipartite. Now since the two extra conditions  $A \subseteq E(H)$  and  $B \cap E(H) = \emptyset$  can be checked in polynomial time, we conclude that  $H$  itself is a polynomial certificate.

We now prove NP-hardness by reduction from 3-SAT. Let  $\varphi = C_1 \wedge \cdots \wedge C_m$  be an instance of 3-SAT on variable set  $V(\varphi) = \{v_1, \dots, v_n\}$ . We construct  $G$  by creating two vertices  $x_i, y_i$  representing the literals of variable  $v_i$  for every  $1 \leq i \leq n$ , a special vertex  $u$ , and a vertex  $c_j$  representing the clause  $C_j$  for every  $1 \leq j \leq m$ . We add the edges  $x_i y_i$  for every  $1 \leq i \leq n$ , the edge  $x_i c_j$  if  $v_i$  appears positively in  $C_j$ , and the edge  $y_i c_j$  if it appears negated. Finally, we connect  $u$  to every other vertex in  $G$  making it a universal vertex. This completes the construction of the graph. Note that it is not bipartite as it contains triangles  $x_i y_i u$  for  $1 \leq i \leq n$ . The two sets  $A, B \subseteq E(G)$  are defined as  $A := \{x_i y_i : 1 \leq i \leq n\}$  and  $B := \{u c_j : 1 \leq j \leq m\}$ . We illustrate the reduction in Figure 5.

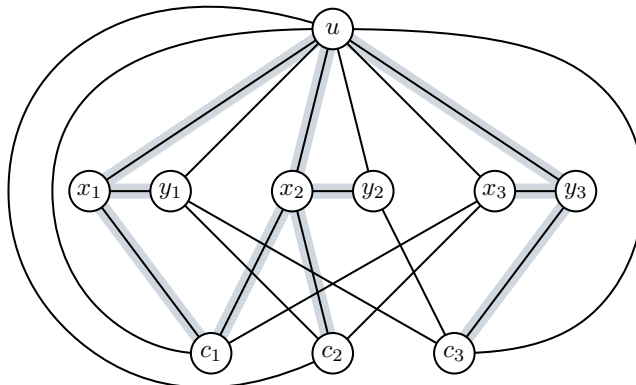


Figure 5: The reduction of Theorem 5.1 with the CNF  $\varphi = (v_1 \vee v_2 \vee v_3) \wedge (\bar{v}_1 \vee v_2 \vee v_3) \wedge (\bar{v}_1 \vee v_2 \vee \bar{v}_3)$ . We have  $A = \{x_1 y_1, x_2 y_2, x_3 y_3\}$  and  $B = \{c_1 u, c_2 u, c_3 u\}$ . A maximal bipartite subgraph including  $A$  and avoiding  $B$  is highlighted in grey. It corresponds to the model of  $\varphi$  which assigns 1 to  $v_1$  and  $v_2$ , and 0 to  $v_3$ .

Let us first show that there exists a maximal bipartite subgraph of  $G$  containing  $A$  and avoiding  $B$  whenever  $\varphi$  is satisfiable. Consider a model  $\mathbf{a}$  of  $\varphi$  and consider the subgraph  $H$  of  $G$  defined as

$$\begin{aligned} E(H) := & A \cup \{x_i u : \mathbf{a}(v_i) = 1\} \cup \{y_i u : \mathbf{a}(v_i) = 0\} \\ & \cup \{c_j x_i : \mathbf{a}(v_i) = 1 \text{ and } v_i \in C_j\} \\ & \cup \{c_j y_i : \mathbf{a}(v_i) = 0 \text{ and } \bar{v}_i \in C_j\}. \end{aligned}$$

Clearly  $H$  extends subset  $A$ . We first argue that it is bipartite. Consider the bipartition  $(V_1, V_2)$  where  $V_1 = \{x_i : \mathbf{a}(v_i) = 1\} \cup \{y_i : \mathbf{a}(v_i) = 0\}$  and  $V_2 = V \setminus V_1$ . By construction  $V_1$  is edgeless. The set  $V_2$  is edgeless too as by the definition of  $H$  vertices adjacent to  $u$  or to  $c_j$ ,  $1 \leq j \leq m$  in  $H$  are precisely those in  $V_1$ . Thus  $H$  is bipartite. We greedily

extend  $H$  into a maximal bipartite subgraph and note  $H'$  the result. Suppose toward a contradiction that  $H'$  does not satisfy the required properties, i.e., that it contains an edge  $uc_j$  for some  $1 \leq j \leq m$ . Then by definition of  $V_1$  and the fact that  $\mathbf{a}$  is a model of  $\varphi$ , we get that  $uc_jw$  would induce a triangle for  $w$  the neighbor of  $c_j$  in  $V_1$ . This contradicts the fact that  $H'$  is bipartite. Consequently  $H'$  extends  $A$  and avoids  $B$  as required.

Suppose now that there exists a maximal bipartite subgraph  $H$  of  $G$  with  $A \subseteq E(H)$  and  $B \cap E(H) = \emptyset$ . Let  $(V_1, V_2)$  be a bipartition of  $H$ . Note that since  $x_1y_1, \dots, x_ny_n$  belong to  $A$  (hence to  $H$ ), the sets  $V_1$  and  $V_2$  contain exactly one and distinct elements from  $\{x_i, y_i\}$  for each  $1 \leq i \leq n$ . Let us assume without loss of generality that  $u \in V_2$ . Then since every edge  $uc_j$ ,  $1 \leq j \leq m$  belongs to  $B$ , it must be that every such  $c_j$  lies in  $V_2$  as well. Thus, except for the choice of which of  $\{x_i, y_i\}$  belongs to  $V_1$ ,  $1 \leq i \leq n$ , the bipartition  $(V_1, V_2)$  is now completely characterized. Consider a triangle  $ux_iy_i$ ,  $1 \leq i \leq n$ . Then one of  $x_i, y_i$  belongs to  $V_1$ , call it  $\ell_1$ , and the other belongs to  $V_2$ , call it  $\ell_2$ . Clearly  $u\ell_2$  is not an edge of  $H$  as both  $u$  and  $\ell_2$  lie in the same side of the bipartition. We argue that  $u\ell_1$  is an edge of  $H$ . Suppose that it is not the case. Then by maximality of  $H$ , adding  $u\ell_1$  creates an odd cycle. Consequently there exists an even path  $P$  going from  $u$  to  $\ell_1$  in  $H$ , where by even we mean that  $P$  contains an even number of edges. Since  $\ell_1\ell_2$  belongs to  $H$  we deduce that the path  $P\ell_2$  obtained by adding  $\ell_2$  to  $P$  is odd, which contradicts the fact that  $u$  and  $\ell_2$  both lie in  $V_2$ . A consequence of these observations is that the set of edges  $u\ell$  in  $H$  for  $\ell \in V_1$  defines an assignment  $\mathbf{a}$  of  $\varphi$ , where  $\mathbf{a}(v_i) = 1$  if  $\ell = x_i$ , and  $\mathbf{a}(v_i) = 0$  if  $\ell = y_i$ . Hence the edges of  $H$  having some  $c_j$  for endpoint are precisely those connecting  $c_j$  to  $V_1$ . We claim that every such  $c_j$  has at least one such incident edge, hence that  $\mathbf{a}$  is a satisfying truth assignment of  $\varphi$ . Indeed, if it was not the case then  $c_j$  would be disconnected from the rest of the graph. But in that case, we could have added  $uc_j$  to the graph still maintaining the bipartiteness (changing  $c_j$  to the other side of the bipartition) contradicting the maximality of  $H$ . This concludes the proof.  $\square$

Concerning prospective research directions, let us restate here the implicit question from [BBČ+21] that motivated this work, which additionally was posed at the WEPA'19 open problem session, and that is still open to date. Recall that a family of CNF's is tractable if there exists a polynomial-time algorithm to decide the satisfiability of any formula (and its subformulas) in the family.

**Question 1.** *Can the maximal signatures of tractable CNF's be enumerated in total-polynomial time?*

Natural examples of tractable CNF's include 2-CNF's and Horn CNF's for which, to the best of our knowledge, no progress has been made. Toward this direction, the case of Horn 2-CNF also seems open.

Another question left open by the work of Bérczi et al. deals with the *dimension* of the formula, defined as the maximum size of a clause it contains. Namely, the algorithm of [BBČ+21, Theorem 4] only performs in incremental-polynomial time for

fixed dimension, and it is open whether the same result can be obtained for formulas of arbitrary dimension.

Finally, another promising direction concerns the parameterized study of signatures enumeration. More particularly, it can be seen that the algorithm of [BBČ+21, Theorem 4] is XP-incremental parameterized by the dimension  $k$ , that is, it generates the  $i^{\text{th}}$  solution in time  $(\|\varphi\| + i)^{f(k)}$  for some computable function  $f$ . It would be interesting to know whether this algorithm can be improved to run with FPT-delay, that is, to run with  $f(k) \cdot \|\varphi\|^{O(1)}$  delay; see e.g. [CMM+17, GKKL22] for more details on the parameterized aspects of enumeration problems. A preliminary step in that direction would be to get such running times for a double parameterization. Among parameters, the maximum number  $\omega$  of occurrence of a variable in the formula is natural. This parameter is relevant as in [BBČ+21, Theorem 3] the authors give a simpler algorithm for signatures enumeration when both  $k$  and  $\omega$  are bounded. Improving this XP-incremental algorithm into one that is FPT-delay parameterized by  $k$  plus  $\omega$  seems open.

**Acknowledgements.** The authors would like to thank Kazuhiro Kurita, Kazuhisa Makino, Kunihiro Wasa, and Yasuaki Kobayashi for preliminary discussions on the topic of this paper.

## References

- [BBČ+21] Kristóf Bérczi, Endre Boros, Ondřej Čepek, Khaled Elbassioni, Petr Kučera, and Kazuhisa Makino. Generating clause sequences of a CNF formula. *Theoretical computer science*, 856:68–74, 2021.
- [BEG04] Endre Boros, Khaled Elbassioni, and Vladimir Gurvich. Algorithms for generating minimal blockers of perfect matchings in bipartite graphs and related problems. In *European Symposium on Algorithms*, pages 122–133. Springer, 2004.
- [BEGK03] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Leonid Khachiyan. Algorithms for enumerating circuits in matroids. In *International Symposium on Algorithms and Computation*, pages 485–494. Springer, 2003.
- [BKPS19] Endre Boros, Benny Kimelfeld, Reinhard Pichler, and Nicole Schweikardt. Enumeration in Data Management (Dagstuhl Seminar 19211). *Dagstuhl Reports*, 9(5):89–109, 2019.
- [CGM+22] Alessio Conte, Roberto Grossi, Andrea Marino, Takeaki Uno, and Luca Versari. Proximity search for maximal subgraph enumeration. *SIAM Journal on Computing*, 51(5):1580–1625, 2022.

- [CH96] Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996.
- [CH97] Nadia Creignou and Jean-Jacques Hébrard. On generating all solutions of generalized satisfiability problems. *RAIRO Theor. Informatics Appl.*, 31(6):499–511, 1997.
- [CH11] Yves Crama and Peter L. Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.
- [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.
- [COS11] Nadia Creignou, Frédéric Olive, and Johannes Schmidt. Enumerating all solutions of a boolean CSP by non-decreasing weight. In Karem A. Sakallah and Laurent Simon, editors, *Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011, Ann Arbor, MI, USA, June 19-22, 2011. Proceedings*, volume 6695 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2011.
- [CU19] Alessio Conte and Takeaki Uno. New polynomial delay bounds for maximal subgraph enumeration by proximity search. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 1179–1190, New York, NY, USA, 2019. Association for Computing Machinery.
- [GKKL22] Petr A. Golovach, Christian Komusiewicz, Dieter Kratsch, and Van Bang Le. Refined notions of parameterized enumeration kernels with applications to matching cut enumeration. *Journal of Computer and System Sciences*, 123:76–102, 2022.
- [JYP88] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.
- [KBB<sup>+</sup>09] Leonid Khachiyan, Endre Boros, Konrad Borys, Vladimir Gurvich, and Khaled Elbassioni. Generating all vertices of a polyhedron is hard. *Twentieth Anniversary Volume: Discrete & Computational Geometry*, pages 1–17, 2009.
- [KBE<sup>+</sup>05] Leonid Khachiyan, Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. On the complexity of some enumeration problems for matroids. *SIAM Journal on Discrete Mathematics*, 19(4):966–984, 2005.

- [MS19] Arnaud Mary and Yann Strozecki. Efficient enumeration of solutions produced by closure operations. *Discret. Math. Theor. Comput. Sci.*, 21(3), 2019.
- [Str19] Yann Strozecki. Enumeration complexity. *Bulletin of EATCS*, 1(129), 2019.
- [TIAS77] Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.