On the Enumeration of Signatures of XOR-CNF's

Nadia Creignou
Aix-Marseille Université, CNRS, LIS, Marseille, France

Oscar Defrain
Aix-Marseille Université, CNRS, LIS, Marseille, France

Frédéric Olive
Aix-Marseille Université, CNRS, LIS, Marseille, France

Simon Vilmin
Aix-Marseille Université, CNRS, LIS, Marseille, France

Abstract

Given a CNF formula φ with clauses $C_1, \ldots, C_m$ over a set of variables V, a truth assignment $\mathbf{a}\colon V \to \{0,1\}$ generates a binary sequence $\sigma_\varphi(\mathbf{a}) = (C_1(\mathbf{a}), \ldots, C_m(\mathbf{a}))$, called a signature of φ, 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-CNF's.

2012 ACM Subject Classification Mathematics of computing → Enumeration; Mathematics of computing → Graph enumeration

Keywords and phrases Algorithmic enumeration, XOR-CNF, signatures, maximal bipartite subgraphs enumeration, extension, proximity search

Digital Object Identifier 10.4230/LIPIcs.WADS.2025.19

Related Version *Full Version*: `arXiv:2402.18537` [9]

Funding The authors have been supported by the ANR project PARADUAL (ANR-24-CE48-0610-01).

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

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 φ with clauses $C_1, \ldots, C_m$ over a set of variables $V = \{v_1, \ldots, v_n\}$, a truth assignment $\mathbf{a}\colon V \to \{0,1\}$ leads to a binary sequence $\sigma_\varphi(\mathbf{a})$, called a *signature* of φ, defined by $\sigma_\varphi(\mathbf{a}) = (C_1(\mathbf{a}), \ldots, C_m(\mathbf{a}))$ 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 φ boils down to deciding whether the all-one sequence is a signature of φ.

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-designed pattern trees and has first been posed during the

19th International Symposium on Algorithms and Data Structures (WADS 2025).
Editors: Pat Morin and Eunjin Oh; Article No. 19; pp. 19:1–19:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Dagstuhl Seminar on enumeration in data management [5]. 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 complexity is not a suitable yardstick of efficiency when analyzing enumeration algorithms performance. Instead, *output-sensitive complexity* estimates the running time of an enumeration algorithm using both input and output sizes. In that framework, an algorithm is considered tractable when it runs in *total-polynomial time* (or *output-polynomial* time), that is, if its execution time is bounded by a polynomial in the combined size of the input and the output. Still, further constraining this notion to guarantee some regularity in the enumeration is sometimes desirable. 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^{th} solution is bounded by a polynomial in the input size plus i (see e.g., [17, 22]).

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.[1] In a recent contribution [2], Bérczi et al. investigate this problem. They give an incremental-polynomial time algorithm for listing all the signatures of a k-CNF for fixed k, and show that for tractable CNF's, signatures can be enumerated with polynomial delay. 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 [23]. 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 φ 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).

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. XOR-clauses were revealed in Schaefer's famous complexity classification as a tractable case for deciding the satisfiability of a set of generalized clauses [21]. Interestingly, XOR-clauses are the only ones that lead to a tractable case when it comes to counting the number of satisfying assignments [11]. They subsequently appeared in many Schaefer-like classifications for a variety of algorithmic tasks, including enumeration [10, 14]. The success of XOR-clauses is due to their algebraic properties. Indeed, a set of such clauses can be seen as a system of linear equations over the 2-element field. As a result, the sets

[1] These problems thus lie outside of the class EnumP which is commonly studied in algorithmic enumeration; see e.g. [22].

of satisfying assignments of such subsets are affine subspaces, and they allow for algebraic tools such as Gaussian elimination and matroid algorithms. Also in the special case of 2-XOR clauses, subsets of clauses can be interpreted as graphs with 0/1-labeled edges and benefit from several efficient graph algorithms, e.g., for computing connected components or identifying cycles [14]. This ability to capture XOR clauses using algebraic, matroid, and graph concepts and methods provides efficient tools for analyzing their behavior with respect to our enumeration problem.

Note that these formulas do not fall in the framework of CNF's though. Indeed, rewritting an 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 [2] do not directly apply to XOR-CNF. Yet, we show that signatures enumeration remains tractable using flashlight search as it is done in [2]. Namely, we prove the following.

▶ **Theorem 1.** *The set of signatures of an 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 2.** *There is polynomial-delay and polynomial space algorithm listing all minimal signatures of an XOR-CNF formula if and only if there is one listing all maximal signatures of an XOR-CNF formula.*

This leaves 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 [6]. 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 [4]; see also [18, 19]. Relying on these results, we derive that maximal signatures enumeration for XOR-CNF's is tractable. Namely, we obtain the following.

▶ **Theorem 3.** *There is an incremental-polynomial time algorithm generating the maximal (or minimal) signatures of an 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 4.** *There is a polynomial-delay algorithm generating the maximal (or minimal) signatures of a 2-XOR-CNF.*

To prove Theorem 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 3 and 4 is that a queue of solutions has to be maintained in the generation. More precisely, in [4] 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 [6], 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 [4] to get polynomial space is open for two decades now. As of improving Theorem 4 to use polynomial space, it would imply such a result for maximal bipartite subgraphs enumeration, another open question [6]. Toward such directions, we prove that the folklore technique of flashlight search used to obtain Theorem 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 and basic properties. However, due to the space constraint, the proofs of the different statements provided in this extended abstract are omitted. The interested reader may still find the arguments in the long version of the manuscript in appendix. The steps to prove the equivalence between maximal and minimal signatures enumeration for XOR-CNF and Theorem 1 are given in Section 3. The paths toward Theorems 3 and 4 are detailed in Section 4. Future directions and the limitations of flashlight search toward an improvement of Theorem 4 are finally discussed in Section 5.

2 Preliminaries

All the objects considered in this paper are finite. If V is a set, $\mathbf{2}^V$ 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^{th} element of σ. It will be furthermore convenient to note $\mathbf{1}(\sigma)$ the set of indices $I \subseteq \{1,\ldots,m\}$ whose corresponding values in σ is 1; $\mathbf{0}(\sigma)$ is defined analogously. The *complementary* of σ is the binary sequence $\bar{\sigma}$ obtained from σ 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 $1 \leq i \leq m$. In other words, $\leq$ corresponds to the bitwise order, and we write $\sigma < \tau$ if $\sigma \leq \tau$ and $\sigma \neq \tau$.

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 [17, 22] 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 write 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 with the same endpoints, i.e., with $E(G) = \{e_1,\ldots,e_m\}$ 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 $V = \{x_1,\ldots,x_n\}$ 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) φ is a conjunction of clauses, i.e., $\varphi = C_1 \wedge \cdots \wedge C_m$ where C_i is a clause for all $1 \leq i \leq m$. It will be convenient to denote by $V(\varphi)$ the variables of φ. The (total) *size* of

φ is defined as $\sum_{1\leq i\leq m}|C_i|$. An assignment of the variables in V is a mapping $\mathbf{a}\colon V\to\{0,1\}$. 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 φ is *satisfiable* if there exists an assignment $\mathbf{a}$ of $V(\varphi)$ such that $\varphi(\mathbf{a})=1$; it is unsatisfiable otherwise.

An *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 $x_1+\cdots+x_k=\varepsilon$, $\varepsilon\in\{0,1\}$ 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. $x_1\neq x_2$). an XOR-clause is *odd* (resp. *even*) if $\varepsilon=1$ (resp. 0). Given an XOR-clause $C=(x_1+\cdots+x_k=\varepsilon)$, with $\varepsilon\in\{0,1\}$, we put $\bar{C}=(x_1+\cdots+x_k=1-\varepsilon)$ and call $\bar{C}$ the *negation* of C. In other words, the bar operator *changes the parity* of C. an XOR-CNF φ is a conjunction of XOR-clauses. Equivalently, φ can be seen as a system of linear equations over $\mathbf{F}_2$. For an XOR-CNF $\varphi=C_1\wedge\cdots\wedge C_m$, we denote by $\bar{\varphi}$ the XOR-CNF $\bar{C}_1\wedge\cdots\wedge\bar{C}_m$ and call it the *inverse* of φ. We recall that the satisfiability of an XOR-CNF can be tested in polynomial time [8, Theorem 2.18].

Signatures. The next notations and terminology are borrowed from [2]. Let $\varphi=C_1\wedge\cdots\wedge C_m$ be a CNF. We refer to the introduction for the definition of a signature, and shall say that an assignment $\mathbf{a}$ of φ *produces* the signature $\sigma_\varphi(\mathbf{a})=(C_1(\mathbf{a}),\ldots,C_m(\mathbf{a}))$. We will further drop the subscript φ from this notation when it is clear from the context. Note that in particular, φ is satisfiable if and only if $(1,\ldots,1)$ is a signature of φ. A signature σ of φ is *minimal* if it is minimal with respect to $\leq$ among all signatures of φ, or in other words, if σ is a minimal element of the partial order induced by $\leq$ on the set of all signatures. Similarly, we call *maximal* a signature which is maximal for $\leq$ among signatures of φ.

In this paper we are interested in the problem of enumerating all (resp. minimal, maximal) signatures of a given XOR-CNF. Observe that in these problems, and by the previous discussions, we may indifferently consider φ 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 φ is in this latter form. In particular, when φ is seen as a system of linear equations over $\mathbf{F}_2$, there is a one-to-one correspondence between maximal signatures of φ 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 an 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 φ may be ignored as far as the enumeration of signatures is concerned. Indeed, having one of these clauses evaluated to 1 implies that each of its copies is also evaluated to 1. 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 an 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 an 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 $V = \{x_1, \ldots, x_n\}$. A preliminary step is the following.

▶ **Proposition 5.** *A sequence $\sigma \in \{0,1\}^m$ is a signature of φ if and only if $\overline{\sigma}$ is a signature of $\bar{\varphi}$. In particular, σ is a minimal (resp. maximal) signature of φ if and only if $\overline{\sigma}$ is a maximal (resp. minimal) signature of $\bar{\varphi}$.*

From Proposition 5 we already derive Proposition 2.

Given two disjoint subsets $A, B \subseteq \{1, \ldots, m\}$, 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 φ 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 φ. Remark that the size of $\varphi(A, B)$ is at most that of φ, and that it is defined on precisely m clauses whenever A, B is a bipartition of $\{1, \ldots, m\}$. In the next proposition, we show that testing whether a binary sequence is a signature of an XOR-CNF can be reduced to satisfiability testing.

▶ **Proposition 6.** *A sequence $\sigma \in \{0,1\}^m$ is a signature of φ if and only if the formula $\varphi(\mathbf{1}(\sigma), \mathbf{0}(\sigma))$ is satisfiable.*

Note that signatures are neither closed by subset or superset, i.e., they may exist binary words $\sigma < \tau < \rho$ such that σ, ρ are signatures but not τ. We nevertheless show that testing the minimality (or maximality) of a signature can be done locally in polynomial time.

▶ **Proposition 7.** *Deciding whether σ is a minimal (resp. maximal) signature of φ can be done in polynomial time in the size of φ.*

To conclude this section, we argue that enumerating all the signatures of an 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 ground set, decide whether there exists a solution including A and avoiding B. We refer to e.g. [3, 20] 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 φ admits a signature σ 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 φ with index in A, and not those with index in B. Thus the signature of φ 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 7 that the extension problem can be solved in polynomial time for XOR-CNF signatures, hence proving Theorem 1 that we recall here.

▶ **Theorem 1.** *The set of signatures of an XOR-CNF formula can be enumerated with polynomial delay and polynomial space.*

We point out that Theorem 1 is not a direct consequence of [2, Theorem 1]. Indeed, in the latter 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. [4], 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 φ may be seen as linear equations $x_1 + \cdots + x_k = \varepsilon$, $\varepsilon \in \{0, 1\}$ over the two-elements field $\mathbf{F}_2$. Hence φ can be seen as a system of linear equations whose maximal feasible subsystems correspond to its maximal signatures. In [4], 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) we can derive 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 [18, 19] on the use of this characterization of infeasible systems. This immediately yields Theorem 3.

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 an XOR-CNF φ 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 φ, and a red edge xy whenever $x = y$ is a clause of φ. 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 φ which typically makes it unsatisfiable. However as no clause is repeated in φ, 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 example of such a multigraph. In the following, we will label edges of G as $e_1, \ldots, e_m$ so that the edge e_j corresponds to the clause C_j in φ. We denote by $R(G)$ the set of red edges of G, and $B(G)$ the set of its blue edges. Observe that by definition, $R(G)$ and $B(G)$ define a bipartition of the edge multiset of G. In other words, it is convenient to see $R(G)$ and $B(G)$ as partitioning the *set* of indices of the edges $e_1, \ldots, e_m$ of G. In the following, when noting $E \subseteq F$ for two edge multisets of G, we mean inclusion of the indices of their edges, that is, by $E \subseteq F$ we mean $I(E) \subseteq I(F)$ for $I(F) \subseteq \{1, \ldots, m\}$ denoting the set of indices of the edges in F.

In the rest of the section, by *subgraph* of an edge-bicolored multigraph G we mean a graph H with $E(H) \subseteq E(G)$ and $V(H) \subseteq V(G)$ where $V(H)$ contains at least the vertices involved in the edges of $E(H)$. For convenience, we may as well refer to a subgraph by its edge set, in which case we assume its vertex set to be the one induced by the endpoints of its edges. Finally, given two disjoint subsets X, Y of a graph G, by $\delta_G(X, Y)$ we mean the set of edges having an endpoint in X and the other in Y, and may drop G from the subscript if no ambiguity arises.

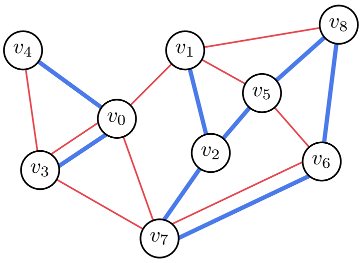

Figure 1 An edge-bicolored multigraph G on nine vertices, with red edges denoted by thin red lines and blue edges denoted by bold blue lines.

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

▶ **Lemma 8.** *Let φ be a graphic 2-XOR-CNF. Then φ 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 φ with $X = \mathbf{a}^{-1}(1)$ and $Y = \mathbf{a}^{-1}(0)$.*

We derive the following analogue for signatures, where by maximal subgraph we refer to edge inclusion.

▶ **Lemma 9.** *Let φ be a graphic 2-XOR-CNF and $\sigma \in \{0,1\}^m$. Then σ is a maximal signature of φ if and only if the subgraph $H = \{e_j : e_j \in E(G),\ \sigma[j] = 1\}$ of $G(\varphi)$ is a maximal subgraph of G admitting a bipartition (X, Y) such that $\delta_H(X, Y) = B(H)$.*

In the following, given an edge-bicolored multigraph G, we call *red-blue bipartite* any of its subgraph H admitting a bipartition (X, Y) of its vertex set such that $\delta(X, Y) = B(H)$. 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.

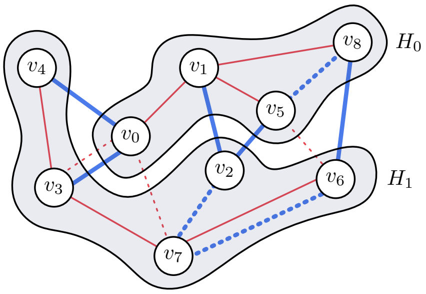

Figure 2 A maximal (connected) red-blue bipartite subgraph H of the multigraph G defined in Figure 1. The bipartition (H_0, H_1) of H is highlighted. The edges of G not in H are dotted.

▶ **Theorem 10.** *There is a polynomial-delay and polynomial-space algorithm enumerating the maximal signatures of a graphic 2-XOR-CNF φ if and only if there is one listing the maximal red-blue bipartite subgraphs of its underlying multigraph $G(\varphi)$.*

A consequence of Theorem 10 is that the enumeration of 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 G is a graph – and we only require $E(G) = \delta(X, Y)$. We derive the following using a recent result by Conte and Uno [6] on maximal bipartite subgraphs enumeration by proximity search.

▶ **Corollary 11.** *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 [6] 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 φ 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 12.** *Let G be a connected edge-bicolored multigraph. Then any of its maximal red-blue bipartite subgraphs is connected.*

▶ **Lemma 13.** *If G is a connected edge-bicolored red-blue 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.*

Proximity search has been introduced in [7] 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 needs not to be symmetric. We refer to [6] for more details on the technique. In [6], 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 14** (Reformulation of [6, Definition 4.2 and Theorem 4.3])**.** *For each $H \in \mathcal{H}(G)$, let $\mu(H)$ denote an ordering of the edges in H and, for arbitrary $H, H^* \in \mathcal{H}(G)$, let $H \tilde{\cap} H^*$ 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 any $H \in \mathcal{H}(G)$, any prefix of $\mu(H)$ is a solution;*
2. *There is a polynomial-time computable function* COMP*, which given a solution H' produces $H = \mathrm{COMP}(H')$ a maximal solution such that $H \supseteq H'$; and*

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 $K \in \mathcal{K}$, $H \setminus K \cup \{e\}$ is a solution; and*
   - *for any $H^* \in \mathcal{H}(G)$, if e is the minimal element of $\mu(H^*)$ not in H, then 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 14 are fulfilled for well-chosen μ 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 greedily obtain a maximal solution.

We start with the definition of μ. Given a maximal red-blue bipartite graph H of G we denote by (H_0, H_1) the unique bipartition given by Lemma 13 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 initiated at v_0: we start with v_0 which is unmarked, and at each step, we iteratively consider unmarked vertices ordered so far, and, in order, we mark them and append the unmarked vertices of their neighborhoods in ascending order of their label. Clearly, each prefix of such an ordering defines a connected (vertex) 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, and from which, intuitively, we will obtain that pairs of solutions H, H^* must agree on their respective bipartition of the elements in $H \tilde{\cap} H^*$.

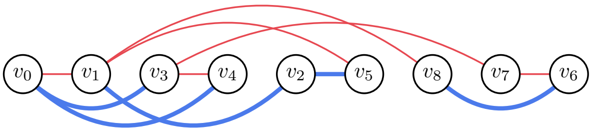

Figure 3 A BFS on the maximal red-blue bipartite subgraph H of G (Figure 2) will give the following order on the vertices: $v_0, v_1, v_3, v_4, v_2, v_5, v_8, v_7, v_6$. The order of the edges of H it produces is $\mu(H) = v_0v_1, v_0v_3, v_0v_4, v_3v_4, v_1v_2, v_1v_5, v_2v_5, v_1v_8, v_3v_7, v_8v_6,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

▶ **Observation 15.** *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 14. 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 15 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 15, H' is a connected red-blue bipartite subgraph of G. Thus by Lemma 13 it admits a unique bipartition (H'_0, H'_1) with blue edges lying between H'_0 and H'_1, and such that $v_0 \in H'_0$. Now since H and H^* contain H' they must agree with that bipartition, meaning that $H'_0 \subseteq H_0 \cap H^*_0$ and $H'_1 \subseteq H_1 \cap H^*_1$, where (H^*_0, H^*_1) denote the unique bipartition of H^* such that $v_0 \in H^*_0$ as provided by Lemma 13. In the following, for each edge $e = ab$ in $E(G) \setminus E(H)$, we use the edges incident to the vertices of e to build the removables. More precisely, 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. We stress the fact that in K_1 or K_2 we may find edges lying in both H and H^*, which is not a problem; rather, we will argue that, when choosing e as the minimal element of $\mu(H^*)$ not in H, one of K_1 or K_2 will be disjoint from $H \tilde{\cap} H^*$, hence that $\mathcal{K}$ defines a valid set of removables.

Consider the first edge $e = ab$ in $\mu(H^*)$ that is not an edge of H. We distinguish two symmetric cases depending 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 $e \notin H$. Since H is maximal, it is connected by Lemma 12, so that a, b lie in $V(H)$. 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 $\{a, b\} \subseteq H_0$, as depicted in Figure 4. 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 arguments, swapping red for blue. Hence, for one of K_1 or K_2 we have that $(H \tilde{\cap} H^*) \cap K = \emptyset$, as required by the last condition of Theorem 14.

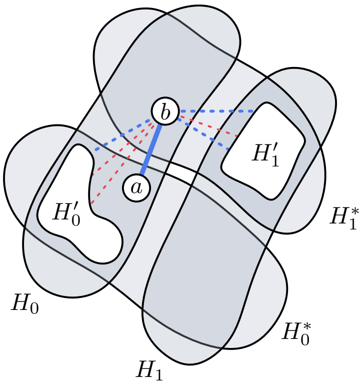

Figure 4 The situation occurring when considering two solutions H, H^* in Theorem 14 with $H' = H \tilde{\cap} H^*$: $H'_0 \subseteq H_0 \cap H_0^*$, $H'_1 \subseteq H_1 \cap H_1^*$ and the blue edge ab of H^* satisfies $E(H') \cap K_2 = \emptyset$ where $K_2 := \{bv : bv \in E(G)\}$.

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 φ does not contain such isolated 1-clauses. We code every other 1-clauses (x) and $(\overline{y})$ – 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 additional vertex u that will be connecting all such 1-clauses. Note that G stays connected by the above assumption that φ 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 4 as a consequence of Theorem 14 and the above discussion.

5 Discussions

In this work we have showed that enumerating all (resp. minimal, maximal) signatures of an XOR-CNF formula is tractable, namely, Theorems 1, 3 and 4. Observe that the algorithm of Theorem 3 runs in incremental-polynomial time while the other two run with polynomial delay. Hence it is natural to ask whether Theorem 3 can be improved to polynomial delay.

▶ **Question 2.** *Can the maximal signatures of an XOR-CNF be listed with polynomial delay?*

The algorithm by Boros et al. [4] underlying Theorem 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 [4] in the Boolean case. A natural first step is to handle the case of k-XOR-CNF's for fixed values of $k \geq 3$.

Furthermore, except for Theorems 1 and 3 that run with polynomial space, the algorithm of Theorem 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 [6].

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 16.** *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.*

Concerning prospective research directions, let us restate here the implicit question from [2] 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 [2, 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 is the parameterized study of signatures enumeration. More particularly, it can be seen that the algorithm of [2, Theorem 4] is XP-incremental parameterized by the dimension k, that is, it generates the i^{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. [1, 12, 13, 15, 16] 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 occurrence ω of a variable in the formula is natural. This parameter is relevant as in [2, Theorem 3] the authors give a simpler algorithm for signatures enumeration when both k and ω are bounded. Improving this XP-incremental algorithm into one that run with FPT-delay parameterized by k plus ω seems open.

References

1 Valentin Bartier, Oscar Defrain, and Fionn Mc Inerney. Hypergraph dualization with FPT-delay parameterized by the degeneracy and dimension. In *International Workshop on Combinatorial Algorithms*, pages 111–125. Springer, 2024. doi:10.1007/978-3-031-63021-7_9.

2 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. doi:10.1016/J.TCS.2020.12.021.

3 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. doi:10.1007/978-3-540-30140-0_13.

4 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. doi:10.1007/978-3-540-24587-2_50.

5 Endre Boros, Benny Kimelfeld, Reinhard Pichler, and Nicole Schweikardt. Enumeration in Data Management (Dagstuhl Seminar 19211). *Dagstuhl Reports*, 9(5):89–109, 2019. doi:10.4230/DagRep.9.5.89.

6 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. doi:10.1137/20M1375048.

7 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. doi:10.1145/3313276.3316402.

8 Yves Crama and Peter L. Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

9 Nadia Creignou, Oscar Defrain, Frédéric Olive, and Simon Vilmin. On the enumeration of signatures of XOR-CNF's. *CoRR*, abs/2402.18537, 2024. doi:10.48550/arXiv.2402.18537.

10 Nadia Creignou and Jean-Jacques Hébrard. On generating all solutions of generalized satisfiability problems. *RAIRO Theor. Informatics Appl.*, 31(6):499–511, 1997. doi:10.1051/ITA/1997310604991.

11 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996. doi:10.1006/INCO.1996.0016.

12 Nadia Creignou, Raïda Ktari, Arne Meier, Julian-Steffen Müller, Frédéric Olive, and Heribert Vollmer. Parameterised enumeration for modification problems. *Algorithms*, 12(9):189, 2019. doi:10.3390/A12090189.

13 Nadia Creignou, Arne Meier, Julian-Steffen Müller, Johannes Schmidt, and Heribert Vollmer. Paradigms for parameterized enumeration. *Theory of Computing Systems*, 60:737–758, 2017. doi:10.1007/S00224-016-9702-4.

14 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. doi:10.1007/978-3-642-21581-0_11.

15 Khaled Elbassioni, Matthias Hagen, and Imran Rauf. Some fixed-parameter tractable classes of hypergraph duality and related problems. In *Parameterized and Exact Computation: Third International Workshop, IWPEC 2008, Victoria, Canada, May 14-16, 2008. Proceedings 3*, pages 91–102. Springer, 2008. doi:10.1007/978-3-540-79723-4_10.

16 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. doi:10.1016/J.JCSS.2021.07.005.

17 David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988. doi:10.1016/0020-0190(88)90065-8.

18 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.

19 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. doi:10.1137/S0895480103428338.

20 Arnaud Mary and Yann Strozecki. Efficient enumeration of solutions produced by closure operations. *Discret. Math. Theor. Comput. Sci.*, 21(3), 2019. doi:10.23638/DMTCS-21-3-22.

21 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proccedings 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978. doi:10.1145/800133.804350.

22 Yann Strozecki. Enumeration complexity. *Bulletin of EATCS*, 1(129), 2019.

23 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. doi:10.1137/0206036.