Technical Report AC-TR-17-009

2017

On the Parameterized Complexity of Finding Small Unsatisfiable Subsets of CNF Formulas and CSP Instances

Ronald de Haan, Iyad Kanj, and
Stefan Szeider

This is the authors' copy of a paper that appears in the ACM Transactions on Computational Logic (TOCL).

www.ac.tuwien.ac.at/tr

On the Parameterized Complexity of Finding Small Unsatisfiable Subsets of CNF Formulas and CSP Instances

Ronald de Haan, Algorithms and Complexity Group, TU Wien, Vienna, Austria
Iyad Kanj, School of Computing, DePaul University, Chicago, IL, United States
Stefan Szeider, Algorithms and Complexity Group, TU Wien, Vienna, Austria

In many practical settings it is useful to find a small unsatisfiable subset of a given unsatisfiable set of constraints. We study this problem from a parameterized complexity perspective, taking the size of the unsatisfiable subset as the natural parameter where the set of constraints is either (i) given as a set of clauses (i.e., a CNF formula), or (ii) as an instance of the constraint satisfaction problem (CSP).

In general, the problem is fixed-parameter *in*tractable. For SAT instances, it was known to be W[1]-complete. We establish A[2]-completeness for CSP instances, where A[2]-hardness prevails already for the Boolean case.

With these fixed-parameter intractability results for the general case in mind, we consider various restricted classes of inputs and draw a detailed complexity landscape. It turns out that often Boolean CSP and CNF formulas behave similarly, but we also identify notable exceptions to this rule.

The main part of this paper is dedicated to classes of inputs that are induced by Boolean constraint languages that Schaefer [1978] identified as the maximal constraint languages with a tractable satisfiability problem. We show that for the CSP setting, the problem of finding small unsatisfiable subsets remains fixed-parameter intractable for all Schaefer languages for which the problem is non-trivial. We show that this is also the case for CNF formulas with the exception of the class of bijunctive (Krom) formulas, which allows for an identification of a small unsatisfiable subset in polynomial time.

In addition, we consider various restricted classes of inputs with bounds on the maximum number of times that a variable occurs (the degree), bounds on the arity of constraints, and bounds on the domain size. For the case of CNF formulas, we show that restricting the degree is enough to obtain fixed-parameter tractability, whereas for the case of CSP instances, one needs to restrict the degree, the arity and the domain size simultaneously to establish fixed-parameter tractability.

Finally, we relate the problem of finding small unsatisfiable subsets of a set of constraints to the problem of identifying whether a given variable-value assignment is entailed or forbidden already by a small subset of constraints. Moreover, we use the connection between the two problems to establish similar parameterized complexity results also for the latter problem.

CCS Concepts: •**Theory of computation** → **Problems, reductions and completeness;** *Constraint and logic programming;*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Parameterized complexity, constraint satisfaction, unsatisfiable subsets, CNF formulas, backbones

ACM Reference Format:
Ronald de Haan, Iyad Kanj, and Stefan Szeider, 2016. On the Parameterized Complexity of Finding Small Unsatisfiable Subsets of CNF Formulas and CSP Instances.

1. INTRODUCTION

In the paradigm of constraint programming, one models a search problem by specifying constraints that solutions must satisfy. Each of these constraints specifies a list of possible variable-value assignments for a subset of variables, and thereby imposes restrictions on the set of solutions. Assignments that simultaneously satisfy all constraints then form the solutions for the search problem. However, when augmenting the set of constraints, one can reach the point where there are no solutions. A set of constraints that has no solutions is called unsatisfiable. A fundamental computational task in the area of constraint programming is to identify unsatisfiable subsets of constraints, if they exist. Moreover, in many settings, it is desirable to find unsatisfiable subsets that are as small as possible—for instance, when refining or modifying an unsatisfiable set of constraints to ensure the existence of solutions. In this paper, we study the problem of finding an unsatisfiable subset of a set of constraints that is of a given maximum size k, from a parameterized complexity point of view.

For every constant k, we can clearly identify all unsatisfiable subsets of size at most k of a set $\mathcal{I}$ of constraints in polynomial time by simply going over all subsets of $\mathcal{I}$ of size at most k. However,

This work is supported by the FWF Austrian Science Fund (Parameterized Compilation, P26200).

if $\mathcal{I}$ consists of m constraints, then this brute-force search requires us to consider m^k subsets, which is impractical already for small values of k (e.g., for $k > 4$). It would be desirable to have an algorithm that finds unsatisfiable subsets of size k in time $f(k) \cdot \|\mathcal{I}\|^d$ where f is a function, $\|\mathcal{I}\|$ denotes the bit-size of the set of constraints, and d is a constant. An algorithm with such a running time would render the problem *fixed-parameter tractable* with respect to parameter k. In order to carry out this multi-variate complexity analysis where we can measure the running time in terms of the size k of the unsatisfiable subset in addition to the input size in bits, we use the framework of parameterized complexity [Cygan et al. 2015; Downey and Fellows 1999; Downey and Fellows 2013; Flum and Grohe 2006; Niedermeier 2006], a field of research that is becoming increasingly popular in the domain of artificial intelligence and constraint satisfaction; see, e.g., [Gaspers and Szeider 2011; Gottlob and Szeider 2006; Samer and Szeider 2010]. Concretely, we consider the following parameterized decision problem.

SMALL-UNSAT-SUBSET
Instance: A set $\mathcal{I}$ of constraints, and a positive integer $k \geq 1$.
Parameter: k.
Question: Is there an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ consisting of k constraints?

Fellows et al. [2006] showed that this problem is already W[1]-hard when the constraints are propositional clauses containing only three literals. The parameterized complexity class W[1] is commonly used to provide theoretical evidence that a problem is not fixed-parameter tractable. Under a common complexity-theoretic assumption, W[1]-hard problems indeed do not admit fixed-parameter tractable algorithms [Chen et al. 2005; Chen et al. 2006; Chen and Kanj 2012]. Thus, the result by Fellows et al. shows that we cannot hope for fixed-parameter tractable algorithms to solve this problem in general (even if we restrict to Boolean domains). Therefore, to develop and improve practical algorithms, it would be useful to identify in what restricted cases the problem becomes fixed-parameter tractable.

Mapping out the (parameterized) complexity landscape of various fragments of a problem can be a valuable first step towards understanding the settings under which different algorithmic approaches could work well in practice. To illustrate this point, consider the work of Crampton, Gutin and Karapetyan [2015], who investigate a problem related to multi-valued workflow satisfiability from a parameterized complexity point of view. This problem is W[1]-hard in the general setting. In [Crampton et al. 2015], they identify a restricted setting of this W[1]-hard problem under which the problem admits a fixed-parameter tractable algorithm. Moreover, they develop an algorithm based on this theoretical result, and show that their algorithm, on instances in this restricted setting, outperforms state-of-the-art algorithmic methods that are based on mixed integer programming. This example shows that obtaining a more detailed theoretical understanding of the computational complexity of a problem can help identify more efficient algorithms for solving the problem.

There are plenty of other examples where theoretical fixed-parameter tractability results have led to significant improvements in practical algorithmic methods. Some celebrated examples of such problems have applications in bioinformatics [Abu-Khzam et al. 2006; Cheetham et al. 2003; Hüffner et al. 2008; Langston et al. 2008; Song et al. 2006]. A noteworthy example is that of the VERTEX COVER problem. A parameterized algorithm for VERTEX COVER was previously implemented by the bioinformatics research group at ETH Zürich to find multiple sequence alignments [Roth-Korostensky 2000; Stege 2000]. Moreover, parameterized algorithms for VERTEX COVER have been implemented—for use in bioinformatics applications—on parallel machines and are quite practical for parameter values up to 400 [Abu-Khzam et al. 2006; Cheetham et al. 2003]. In general, theoretical results establishing the parameterized intractability of a problem that has important applications may lead to the identification of other meaningful parameterizations, with respect to which the problem becomes fixed-parameter tractable. This, in turn, may lead to the design of efficient fixed-parameter tractable algorithms for the problem with respect to the newly-identified parameterizations (see, e.g., [Song et al. 2006]).

In this paper, we study the parameterized complexity of SMALL-UNSAT-SUBSET for various restricted classes of constraints. In this investigation, we establish fixed-parameter tractability results for several classes of constraints, as well as negative results indicating that fixed-parameter tractability is not possible for several classes of constraints. These negative results consist of hardness (or completeness) results for various parameterized intractability classes, such as W[1], co-W[1], W[2] and A[2]. Whenever a parameterized problem is hard for any of these classes, it does not admit a fixed-parameter tractable algorithm, unless 3SAT can be solved in subexponential time [Chen et al. 2005; Chen et al. 2006; Chen and Kanj 2012].

1.1. Contributions

Our main results consist of a parameterized complexity classification for the problem SMALL-UNSAT-SUBSET for two general types of constraint formalisms: (1) CNF formulas and (2) CSP instances. We denote the corresponding decision problems by SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET, respectively. For both settings, we study various classes of instances. For an overview of this parameterized complexity classification, see Tables I and II.

Firstly, for both settings, we consider the unrestricted case. The problem SMALL-CNF-UNSAT-SUBSET was shown to be W[1]-complete by Fellows et al. [2006]. The problem SMALL-CSP-UNSAT-SUBSET is harder.

— We show that SMALL-CSP-UNSAT-SUBSET is A[2]-complete (Theorem 3.10).
— Moreover, we show that it is A[2]-hard even when restricted to a Boolean domain (Corollary 3.8).

It is worth pointing out that this A[2]-completeness result is of independent interest, since to the best of our knowledge, this is the first natural problem—that originates in a setting not directly related to the structural development of the A-hierarchy—that is complete for A[2].

Then, both for CNF formulas and for Boolean CSP instances, we consider a number of constraint languages. A constraint language is a set of constraints, and each constraint language naturally induces a class of instances (namely those instances containing only constraints in the constraint language). In particular, we consider Schaefer's [1978] constraint languages, which are the maximal constraint languages that admit a polynomial-time satisfiability check: (i) the language of all 0-valid constraints, (ii) the language of all 1-valid constraints, (iii) the language of all Horn constraints, (iv) the language of all anti-Horn constraints, (v) the language of all bijunctive constraints, and (vi) the language of all affine constraints.

Interestingly, the problem of identifying small unsatisfiable subsets of CSP instances is fixed-parameter *in*tractable (W[1]-hard or harder) for all constraint languages that we consider—for which the problem is non-trivial. Since we consider constraint languages for which deciding satisfiability is tractable, this suggests that the selection of a small subset of constraints comprises a source of complexity by itself. Moreover, in all cases, compared to the setting with CNF formulas, the problem is at least as hard (and in many cases harder) when dealing with CSP instances. It is worth mentioning that these increases in complexity are not due to the domain size, since in both cases the domain is Boolean.

The problem of finding small unsatisfiable subsets is trivially tractable when restricted to 0-valid or 1-valid constraint languages (Observation 4.1). For Horn and anti-Horn constraints, the problem was already shown to be W[1]-hard by Fellows et al. [2006]. We show that W[1]-hardness holds even for two more restricted fragments.

— We show that SMALL-CNF-UNSAT-SUBSET is W[1]-hard even when restricted to instances with only Horn constraints of arity at most 3, where one constraint is a unit clause with a negative literal, and all other constraints are definite Horn clauses (Proposition 4.4),
— We show that SMALL-CNF-UNSAT-SUBSET is W[1]-hard even when restricted to instances with only Horn constraints of arity at most 3 and with only a single unit clause (Corollary 4.6).

These results directly give us W[1]-hardness for the case of CSP instances with bounded arity. In the case of CSP instances of unbounded arity, we even get hardness for W[2].

— We show that SMALL-CSP-UNSAT-SUBSET is W[2]-hard when restricted to instances containing only Horn constraints of arbitrary arity (Proposition 4.7).

The above hardness results (both for the case of CNF formulas and CSP instances) also extend to analogous restrictions for anti-Horn constraints. For bijunctive constraints, we show that the problem increases in complexity when moving from CNF formulas to CSP instances—in the former case, the problem is polynomial-time solvable [Buresh-Oppenheim and Mitchell 2006].

— We show that SMALL-CSP-UNSAT-SUBSET is W[1]-hard when restricted to bijunctive Boolean CSP instances of bounded arity (Proposition 4.9).
— We show that SMALL-CSP-UNSAT-SUBSET is W[2]-hard when restricted to bijunctive Boolean CSP instances of unbounded arity (Corollary 4.10).

For affine constraints, the problem is W[1]-hard both in the setting of propositional formulas and in the setting of CSP instances.

— We show that SMALL-CNF-UNSAT-SUBSET is W[1]-hard when restricted to affine formulas (Proposition 4.12).
— We show that SMALL-CSP-UNSAT-SUBSET is W[1]-hard when restricted to affine Boolean CSP instances (Corollary 4.11).

We further investigate various classes of instances with bounds on (combinations of) the following: (i) the maximum arity of constraints, (ii) the maximum number of times that any variable occurs in the set of constraints (the degree), and (iii) the domain size. In the case of CNF formulas, the problem of deciding whether an instance has an unsatisfiable subset of size k can be done in fixed-parameter tractable time, when the degree of the instance is bounded by a function of k. This result was already discovered by Fellows et al. [2006], using a meta-theorem.

— We give a direct algorithm to solve this problem in fixed-parameter linear time (Proposition 5.1).

For the case of CSP instances, we show that bounding only two of (i–iii) at a time does not lead to fixed-parameter tractability. When restricted to instances with maximum arity 3 and domain size 2, the problem is known to be W[1]-hard (Proposition 3.1, [Fellows et al. 2006]).

— We show that the problem SMALL-CSP-UNSAT-SUBSET is co-W[1]-hard when restricted to instances with maximum arity 2 and degree 3 (Corollary 5.4).
— We show that the problem SMALL-CSP-UNSAT-SUBSET is W[1]-hard when restricted to instances with degree 2 and domain size 2 (Proposition 5.2).

However, when the maximum arity, the degree, and the domain size are all bounded by a function of the parameter value k, the problem of finding unsatisfiable subsets of size k becomes fixed-parameter tractable (Proposition 5.5).

Since A[2] is a parameterized analogue of the classical complexity class Σ_2^p, the A[2]-completeness result for the general case in the setting of CSP instances suggests that the unparameterized variant of SMALL-CSP-UNSAT-SUBSET is Σ_2^p-complete. For the case of CNF formulas, such a Σ_2^p-completeness result was already known (Proposition 3.12, [Liberatore 2005]).

— We show that for the case of CSP instances, the unparameterized variant of the problem is also Σ_2^p-complete (Proposition 3.13).

Finally, we consider the closely related parameterized problem of deciding whether a small subset of constraints already enforces a given variable-value assignment.

restriction	SMALL-CNF-UNSAT-SUBSET		SMALL-CSP-UNSAT-SUBSET[arity]		SMALL-CSP-UNSAT-SUBSET	
unrestricted	W[1]-c		W[2]-h, co-W[1]-h, in A[2]	(Prop 3.2, 3.9, Cor 5.4)	A[2]-c	(Thm 3.10)
Boolean			W[1]-c	(Prop 3.1, 3.16)	A[2]-c	(Thm 3.10)
0-valid, 1-valid	trivial	(Obs 4.1)	trivial	(Obs 4.1)	trivial	(Obs 4.1)
Horn, anti-Horn	W[1]-c		W[1]-c	(Prop 3.16, Cor 4.5)	W[2]-h, in A[2]	(Prop 3.9, 4.7)
bijunctive (Krom)	polynomial time		W[1]-c	(Prop 3.16, 4.9)	W[2]-h, in A[2]	(Prop 3.9, 4.10)
affine	W[1]-h	(Prop 4.12)	W[1]-c	(Prop 3.16, Cor 4.11)	W[1]-h, in A[2]	(Prop 3.9, Cor 4.11)

Table I: Map of parameterized complexity results for the problems SMALL-CNF-UNSAT-SUBSET, SMALL-CSP-UNSAT-SUBSET[arity] and SMALL-CSP-UNSAT-SUBSET, for various restricted classes of (Boolean) instances. The problem SMALL-CSP-UNSAT-SUBSET[arity] is the variant of SMALL-CSP-UNSAT-SUBSET where the problem is additionally parameterized by the maximum arity of the constraints in the input.

restriction	SMALL-CNF-UNSAT-SUBSET		SMALL-CSP-UNSAT-SUBSET	
small arity & domain	W[1]-c		W[1]-c	(Prop 3.16)
small arity & degree	FPT	(Prop 5.1)	co-W[1]-h, in A[2]	(Prop 3.9, Cor 5.4)
small degree & domain	FPT	(Prop 5.1)	W[1]-h, in A[2]	(Prop 3.9, 5.2)
small arity, degree & domain	FPT	(Prop 5.1)	FPT	(Prop 5.5)

Table II: Map of parameterized complexity results for the problems SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET, for various combinations of restrictions on the arity, degree and domain size.

— We show that both for the case of CNF formulas and for the case of CSP instances, this problem is fpt-reducible to the problem of finding small unsatisfiable subsets (Lemmas 6.2 and 6.8), and vice versa (Lemmas 6.1 and 6.7).

Moreover, this interreducibility holds for almost all of the restricted classes of instances that we consider. The only exceptions are the classes of 0-valid and 1-valid CNF formulas and CSP instances. The problem of finding small unsatisfiable subsets is trivial for 0-valid and 1-valid CNF formulas and CSP instances.

— We show that the problem of identifying local backbones is in fact W[1]-complete for CNF formulas (Proposition 6.3) and A[2]-complete for CSP instances (Proposition 6.9).

1.2. Case Study: Simplifying Formulas

To highlight the relevance of the computational task of identifying small unsatisfiable subsets, we consider one scenario where efficient algorithms for this task would be very beneficial. In this scenario, we simplify propositional CNF formulas by identifying backbones. A *backbone* is a variable whose truth value is the same for all satisfying assignments. If a backbone variable and its corresponding truth value are known, then we can assign this value to the variable, and thereby simplify the formula without changing its satisfiability or the set of satisfying assignments for this formula. Unfortunately, the problem of identifying backbones is as hard as finding unsatisfiable subsets (a co-NP-complete problem)—that is, identifying backbones is co-NP-complete as well—and therefore, cannot be solved efficiently in general.

However, a variable can be a backbone because of *local properties* of the formula, that is, it is a backbone of a small subset of the constraints. Such backbones we call *local backbones*. As an extreme example consider a CNF formula that contains a unit clause. In this case we know that the variable appearing in the unit clause is a backbone of the formula. The problems of finding local backbones and finding small unsatisfiable subsets are closely related. In particular, we can employ algorithms that identify small unsatisfiable subsets to find local backbones (and their corresponding truth value)—we refer to Section 6 for more details.

The motivation for studying local backbones is that local backbones are a particular type of backbones that—in some cases—can be identified more efficiently. To give a concrete example, we preview a result that we will establish in Sections 5 and 6, and we consider the setting of CNF formulas of degree 3—i.e., each variable occurs at most three times. In this setting, the satisfiability problem remains NP-complete, and as a result, deciding whether a given variable is a backbone remains co-NP-complete. However, as we will show, in this setting local backbones can be computed in fixed-parameter tractable time with respect to the order k of the local backbone (Lemma 6.2 and Proposition 5.1).

After instantiating a local backbone variable with its corresponding truth value, there might be new local backbone variables. Such backbone variables that can be identified by repeatedly finding local backbones and simplifying the set of constraints we call *iterative local backbones*.

To quantify to what extent local backbones and iterative local backbones are really local, we consider the following notion of (iterative) order for backbones. Let x be a backbone of a CNF formula φ. We say that the *order* of x is the cardinality of a smallest subset $\varphi' \subseteq \varphi$ such that x is a backbone of φ'. Similarly, we say that the *iterative order* of x is the smallest number k such that x can be identified as a backbone of φ by repeatedly finding backbones of order k, instantiating them with their corresponding truth value and thusly simplifying the formula φ.

As an indication of the potential of the technique for finding backbones iteratively in this way, by finding and instantiating local backbones, we provide some experimental results in Appendix A, which show the low (iterative) order of many backbones in several SAT instances from various domains. These results thus illustrate that efficient algorithms for identifying small unsatisfiable subsets could be very useful for simplifying constraint instances from various practical domains.

1.3. Related Work

There has been much research on the topic of developing fast algorithms to find minimal (subset-minimal) or minimum (cardinality-minimal) unsatisfiable subsets for propositional formulas, e.g., [Bacchus and Katsirelos 2015; Belov et al. 2012; Ignatiev et al. 2015; Lynce and Silva 2004; Marques-Silva 2012], possibly over an underlying theory (SMT) [Cimatti et al. 2011], as well as for instances of the Constraint Satisfaction Problem (CSP) [Hemery et al. 2006]. The problem of identifying unsatisfiable subsets of size k has been considered from a parameterized complexity point of view by Fellows, Szeider and Wrightson [2006], who proved that this problem (parameterized by k) is W[1]-complete. Furthermore, they showed by the same reduction that finding a k-step resolution refutation for a given formula is W[1]-complete as well.

The notion of backbones has initially been studied in the context of optimization problems in computational physics [Schneider et al. 1996]. Backbones have also been considered in other contexts, such as knowledge compilation [Darwiche and Marquis 2002], and for other combinatorial problems [Slaney and Walsh 2001], including SAT. The relation between backbones and the difficulty of finding a solution for SAT has been studied before [Kilby et al. 2005; Parkes 1997; Slaney and Walsh 2001]. Moreover, the notion of backbones has been used for improving SAT solving algorithms [Dubois and Dequen 2001; Hertli et al. 2011]. Related notions of locally enforced literals have also been studied, including a notion of generalized unit-refutation [Gwynne and Kullmann 2013; Kullmann 1999].

This paper directly extends the research of Fellows et al. [2006]. Preliminary results have appeared in conference proceedings [De Haan et al. 2013a; De Haan et al. 2014] and in a technical report [De Haan et al. 2013b].

1.4. Roadmap

We begin in Section 2 with reviewing relevant definitions from (parameterized) complexity theory, propositional satisfiability and constraint satisfaction. Then, in Section 3, we consider the parameterized complexity of the problem of finding small unsatisfiable subsets, in the general case and for the case of Boolean constraints, both for CNF formulas and CSP instances. In Section 4, we investigate the problem for several classes of Boolean constraints for which the satisfiability problem is polynomial-time solvable. Then, in Section 5, we consider various classes of instances where each variable can occur only a bounded number of times. In Section 6, we relate the problem of finding small unsatisfiable subsets to the problem of identifying local backbones. Finally, we conclude in Section 7.

2. PRELIMINARIES

Before we can begin our parameterized complexity analysis of the problem of finding small unsatisfiable subsets, we briefly review the relevant concepts and tools from (parameterized) complexity theory, propositional satisfiability and constraint satisfaction. Moreover, we formally define the parameterized problems SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET.

2.1. (Parameterized) Complexity

We begin with introducing the relevant concepts of parameterized complexity theory. For more details, we refer to textbooks on the topic [Cygan et al. 2015; Downey and Fellows 1999; Downey and Fellows 2013; Flum and Grohe 2006; Niedermeier 2006]. An instance of a parameterized problem is a pair (I, k) where I is the main part of the instance, and k is the parameter. A parameterized problem is *fixed-parameter tractable* if instances (I, k) of the problem can be solved by a deterministic algorithm that runs in time $f(k) \cdot \|I\|^c$, where f is a computable function of k, c is a constant, and $\|I\|$ denotes the bit-size of the instance I (algorithms running within such time bounds are called *fpt-algorithms*). If $c = 1$, we say the problem is *fixed-parameter linear*. FPT denotes the class of all fixed-parameter tractable problems. Using fixed-parameter tractability, many problems that are classified as intractable in the classical setting (i.e., NP-hard) can be shown to be tractable for small values of the parameter.

Parameterized complexity also offers a *completeness theory*, similar to the theory of NP-completeness, that provides a way to obtain strong theoretical evidence that a parameterized problem is not fixed-parameter tractable by showing that a parameterized problem is hard (or complete) for one of various *parameterized intractability classes*. Hardness for parameterized complexity classes is based on fpt-reductions, which are many-one reductions where the parameter of one problem maps into the parameter for the other. More specifically, a parameterized problem L is fpt-reducible to another parameterized problem L' if there is a mapping R that maps instances of L to instances of L' such that (i) $(I, k) \in L$ if and only if $R(I, k) = (I', k') \in L'$, (ii) $k' \leq g(k)$ for a computable function g, and (iii) R can be computed in time $f(k) \cdot \|I\|^c$ for a computable function f and a constant c.

The intractability class XP includes all *xp-solvable* parameterized problems, which are those parameterized problems that can be solved by an *xp-algorithm*, i.e., an algorithm with running time $O(n^{f(k)})$, for some computable function f, where n is the input size and k is the parameter value.

Central to the completeness theory is the W-hierarchy consisting of the intractability classes W[t]: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ W[P] $\subseteq$ XP. The parameterized complexity classes W[t] for $t \geq 1$ and W[P] are based on the weighted satisfiability problems for Boolean circuits. We consider *Boolean circuits* with a single output gate. Boolean circuits are directed acyclic graphs, where each node with no ingoing edges is called an input node (or a *variable*), and where all other nodes are labelled with a Boolean operator (and are called *gates*). If there is an edge from a node r to a node r', we say that r is an *input* (or a *parent*) of r'. Gates that are labelled with a negation have exactly one input, and gates that are labelled with conjunction or negation can have more inputs. The number of inputs of a gate is called the *fan-in* of that gate. Similarly, the *fan-out* of a gate is the number of gates that have that gate as input. We distinguish between *small* gates, with fan-in at most 2, and *large gates*, with fan-in greater than 2. The *depth* of a circuit is the length of a longest path from any variable to the output gate. The *weft* of a circuit is the largest number of large gates on any path from a variable to the output gate. We adopt the usual notions of truth assignments and satisfiability of a Boolean circuit. We say that a truth assignment for a Boolean circuit has *weight* k if it sets exactly k of the variables of the circuit to true. We denote the class of Boolean circuits with depth u and weft t by $\textsc{circ}_{t,u}$, and we denote the class of all Boolean circuits by $\textsc{circ}$. For any class $\mathcal{C}$ of Boolean circuits, we define the following parameterized problem.

WSAT[$\mathcal{C}$]
Instance: A Boolean circuit $C \in \mathcal{C}$, and an integer k.
Parameter: k.
Question: Does there exist an assignment of weight k that satisfies C?

For each $t \geq 1$, the parameterized complexity class W[t] consists of all parameterized problems that are fpt-reducible to WSAT[$\textsc{circ}_{t,u}$], for some fixed $u \geq 1$. Similarly, the class W[P] consists of all parameterized problems that are fpt-reducible to WSAT[$\textsc{circ}$].

In addition, the completeness theory of parameterized complexity contains the A-hierarchy, containing the intractability classes A[t], for $t \geq 1$. These classes are based on model checking problems for first-order logic. A *signature* τ is a set of relation symbols R, each associated with an arity a_R. A *structure* over the signature τ consists of a set A called the *universe*, and an interpretation $R^{\mathcal{A}} \subseteq A^{a_R}$ of each relation symbol R of τ. A *first-order logic formula* is a formula built using existential quantification ($\exists x$), universal quantification ($\forall x$), atoms ($R(x_1, \dots, x_{a_R})$), and Boolean connectives ($\wedge, \vee, \neg$). For more details, we refer to textbooks (see, e.g., [Flum and Grohe 2006, Section 4.2]). A *first-order logic sentence* is a first-order logic formula that contains no free variables, i.e., where each variable is bound by a quantifier. A first-order logic sentence is *positive* if it contains only existential quantification and uses only the Boolean operators $\wedge$ and $\vee$. For each $t \geq 1$, the class A[t] is defined as the class of all parameterized problems that are fpt-reducible to the following parameterized problem $\mathrm{MC}(\Sigma_t)$.

$\mathrm{MC}(\Sigma_t)$
Instance: A structure $\mathcal{A}$ (over a signature τ), and a first-order logic sentence $\varphi = \exists x_{1,1}, \dots, x_{1,\ell_1}.\forall x_{2,1}, \dots, x_{x,\ell_2} \dots Q_t x_{t,1}, \dots, x_{t,\ell_t}.\psi$ (over τ), where $Q_t = \exists$ if t is odd and $Q_t = \forall$ if t is even, and where ψ is quantifier-free.
Parameter: $|\varphi|$.
Question: $\mathcal{A} \models \varphi$?

The class A[1] coincides with W[1], and for each $t \geq 2$ it holds that W[t] $\subseteq$ A[t] $\subseteq$ A[$t+1$] $\subseteq \dots \subseteq$ XP. For the case of A[2], the problem $\mathrm{MC}(\Sigma_2)$ remains hard when (1) $\ell_1 = \ell_2$, (2) τ is a fixed signature containing only binary predicates, and (3) ψ is a disjunction of atoms [Flum and Grohe 2006, Lemma 8.10].

The fixed-parameter tractability of a problem that is hard for any of these parameterized intractability classes is unlikely, as it would violate commonly-believed assumptions in complexity theory, such as the Exponential Time Hypothesis (i.e., the existence of a $2^{o(n)}$-time algorithm for n-variable 3SAT) [Chen et al. 2005; Chen et al. 2006; Chen and Kanj 2012; Impagliazzo et al. 2001].

In this paper, we will use the following problems to prove fixed-parameter intractability results. CLIQUE is a W[1]-complete problem [Downey and Fellows 1995b]. The instances are tuples (V, E, k), where (V, E) is a simple graph, and $k \geq 1$ is a positive integer. The parameter is k. The question is whether there exists a k-clique in (V, E).

MULTI-COLORED-CLIQUE is a W[1]-complete problem [Fellows et al. 2009]. The instances are tuples (V, E, k), where V is a finite set of vertices partitioned into k subsets $V_1, \dots, V_k$, (V, E) is a simple graph, and k is a positive integer. The parameter is k. The question is whether there exists a k-clique in (V, E) that contains a vertex in each V_i.

HITTING-SET is a W[2]-complete problem [Downey and Fellows 1995a]. The instances are tuples $(U, \mathcal{T}, k)$, where U is a finite universe, $\mathcal{T}$ is a collection of subsets of U, and $1 \leq k \leq |U|$ is a positive integer. The parameter is k. The question is whether there exists a hitting set $H \subseteq U$ such that $|H| \leq k$ and $H \cap T \neq \emptyset$ for all $T \in \mathcal{T}$.

The problem MC(positive) is W[1]-complete [Papadimitriou and Yannakakis 1999]. Instances of this problem consist of a first-order structure $\mathcal{A}$ (over a signature τ), and a positive first-order logic sentence φ (over the same signature τ). The parameter is $|\varphi|$, and the question is to decide whether $\mathcal{A} \models \varphi$.

2.1.1. The Polynomial Hierarchy. In addition, we need to introduce a few notions from classical complexity theory. We assume that the reader is familiar with the complexity classes P and NP (for an introduction to these classes, we refer to textbooks, e.g., [Arora and Barak 2009]). There are many natural decision problems that are not contained in the classes P and NP. The *Polynomial Hierarchy (PH)* [Meyer and Stockmeyer 1972; Papadimitriou 1994; Stockmeyer 1976; Wrathall 1976] contains a hierarchy of increasing complexity classes Σ_i^{p}, for all $i \geq 0$. We give a characterization of these classes based on the satisfiability problem of various classes of quantified Boolean formulas. A *quantified Boolean formula* is a formula of the form $Q_1 X_1 Q_2 X_2 \dots Q_m X_m \psi$, where each Q_i is either $\forall$ or $\exists$, the X_i are disjoint sets of propositional variables, and ψ is a Boolean formula over the variables in $\bigcup_{i=1}^{m} X_i$. The quantifier-free part of such formulas is called the *matrix* of the formula. Truth of such formulas is defined in the usual way. For each $i \geq 1$ we define the following decision problem.

QSAT$_i$
Instance: A quantified Boolean formula $\varphi = \exists X_1 \forall X_2 \exists X_3 \dots Q_i X_i\, \psi$, where Q_i is a universal quantifier if i is even and an existential quantifier if i is odd.
Question: Is φ true?

For each nonnegative integer $i \leq 0$, the complexity class Σ_i^{p} can be characterized as the closure of the problem QSAT$_i$ under polynomial-time reductions [Stockmeyer 1976; Wrathall 1976]. The Σ_i^{p}-hardness of QSAT$_i$ holds already when the matrix of the input formula is restricted to 3CNF for odd i, and restricted to 3DNF for even i. The class Σ_0^{p} coincides with P, and the class Σ_1^{p} coincides with NP.

2.2. Propositional Logic

A *literal* is a propositional variable x or a negated variable $\neg x$. A *clause* is a finite set of literals, not containing a complementary pair x, $\neg x$, and unless stated otherwise, it is interpreted as the disjunction of these literals. A *CNF formula* is a finite set of clauses, and is interpreted as the conjunction of these clauses.

A CNF formula φ is a k-CNF formula if the size of each of its clauses is at most k. A 2-CNF formula is also called a Krom formula, or a bijunctive formula. A clause is a *Horn clause* if it contains at most one positive literal. A clause is a *definite Horn clause* if it contains exactly one positive literal. CNF formulas containing only Horn clauses are called *Horn formulas*. CNF formulas containing only definite Horn clauses are called *definite Horn formulas*. A clause is an *anti-Horn clause* if it contains at most one negative literal. CNF formulas containing only anti-Horn clauses are called *anti-Horn formulas*. A CNF formula is 0-*valid* if each clause contains at least one negative literal, and 1-*valid* if each clause contains at least one positive literal.

The *degree* of a propositional variable x in CNF formula φ is the number of clauses of φ in which it occurs (positively or negatively). The degree of φ is the maximum degree of any variable that occurs in φ. We say that a class of CNF formulas has *bounded degree* if there exists a constant $d \geq 1$ such that each formula in the class has degree at most d.

A CNF formula φ is *satisfiable* if there exists a truth assignment $\tau : \mathrm{Var}(\varphi) \to \{0,1\}$ such that every clause $c \in \varphi$ contains some literal l such that $\tau(l) = 1$ (we say that such an assigment τ satisfies φ); otherwise, φ is *unsatisfiable*.

An *affine clause* is a finite set of literals, not containing a complementary pair x, $\neg x$, and is interpreted as the exclusive disjunction (denoted by the symbol $\oplus$) of these literals. That is, an affine clause is true if and only if an odd number of literals appearing in the clause are true. An *affine* formula is a finite set of affine clauses, and is interpreted as the conjunction of these clauses. An affine formula φ is a k-affine formula if the size of each of its affine clauses is at most k.

We say that a CNF or affine formula φ containing variables $x_1, \dots, x_n$ is equivalent to a Boolean relation $R \subseteq \{0,1\}^n$ if the set of assignments to the variables $x_1, \dots, x_n$ that satisfy φ corresponds exactly to the tuples in R.

It is well-known that any minimal unsatisfiable CNF formula has more clauses than variables (this is known as Tarsi's Lemma [Aharoni and Linial 1986; Kullmann 2000a]).

For two formulas φ, ψ, whenever all assignments satisfying φ also satisfy ψ, we write $\varphi \models \psi$. The reduct $\varphi|_L$ of a formula φ with respect to a set of literals $L \subseteq \mathrm{Lit}(\varphi)$ is the set of clauses of φ that do not contain any $l \in L$ with all occurrences of $\overline{l}$ for all $l \in L$ removed. For singletons $L = \{l\}$, we also write $\varphi|_l$. We say that a class $\mathcal{C}$ of formulas is *closed under variable instantiation* if for every $\varphi \in \mathcal{C}$ and every $l \in \mathrm{Lit}(\varphi)$ we have that $\varphi|_l \in \mathcal{C}$. For an integer k, a variable x is a k-*backbone* of φ, if there exists a $\varphi' \subseteq \varphi$ such that $|\varphi'| \leq k$ and either $\varphi' \models x$ or $\varphi' \models \neg x$. A variable x is a *backbone* of a formula φ if it is a $|\varphi|$-backbone. Note that the definition of the backbone of a formula φ that is used in some of the literature includes all literals $l \in \mathrm{Lit}(\varphi)$ such that $\varphi \models l$. For an integer k, a variable x is an *iterative k-backbone* of φ if either (i) x is a k-backbone of φ, or (ii) there exists $y \in \text{Var}(\varphi)$ such that y is a k-backbone of φ, and for some $l \in \{y, \neg y\}$, $\varphi \models l$ and x is an iterative k-backbone of $\varphi|_l$.

For a Krom formula φ, we let $\text{impl}(\varphi)$ be the *implication graph* (V, E) of φ, where $V = \{ x, \neg x : x \in \text{Var}(\varphi) \}$ and $E = \{ (\overline{a}, b), (\overline{b}, a) : \{a, b\} \in \varphi \}$. We say that a path p in this graph *uses a clause* $\{a, b\}$ of φ if either one of the edges $(\overline{a}, b)$ and $(\overline{b}, a)$ occurs in p; we say that p *doubly uses* this clause if both edges occur in p.

The parameterized problem of deciding whether a given formula has an unsatisfiable subset of size k we denote by SMALL-CNF-UNSAT-SUBSET. When considering affine formulas (instead of CNF formulas), we slightly abuse notation, and denote the analogous problem where the input is an affine formula also by SMALL-CNF-UNSAT-SUBSET.

SMALL-CNF-UNSAT-SUBSET
Instance: A CNF formula φ, and a positive integer $k \geq 1$.
Parameter: k.
Question: Is there an unsatisfiable subset $\varphi' \subseteq \varphi$ consisting of k clauses?

2.3. Constraint Satisfaction

Let D be a finite set of values (called the *domain*). An *n-ary relation* on D is a set of n-tuples of elements from D; we use $\mathbf{R}_D$ to denote the set of all relations on D with finite arity. A *constraint language* is a subset of $\mathbf{R}_D$.

Let $\mathcal{V}$ be an infinite set of variables. A *constraint (over a constraint language* $\Gamma \subseteq \mathbf{R}_D$*) of arity* n is a pair (S, R) where $S = (v_1, \dots, v_n)$ is a sequence of variables from $\mathcal{V}$ and $R \in \Gamma$ is a relation in the constraint language Γ (called the *constraint relation*). The set $\text{Var}(C) = \{v_1, \dots, v_n\}$ is called the *scope* of C. An *assignment* $\alpha : V \to D$ is a mapping defined on a set $V \subseteq \mathcal{V}$ of variables. An assigment $\alpha : V \to D$ *satisfies* a constraint $C = ((v_1, \dots, v_n), R)$ if $\text{Var}(C) \subseteq V$ and $(\alpha(v_1), \dots, \alpha(v_n)) \in R$. For a set $\mathcal{I}$ of constraints, we write $\text{Var}(\mathcal{I}) = \bigcup_{C \in \mathcal{I}} \text{Var}(C)$, and we write $\text{Rel}(\mathcal{I}) = \{ R : (S, R) \in C, C \in \mathcal{I} \}$. If the domain D is not explicitly given, we can derive it from any set $\mathcal{I}$ of constraints by taking the set of all values occurring in the constraint relation of any constraint in $\mathcal{I}$.

For any variable $v \in \text{Var}(\mathcal{I})$, we define the *degree* of v to be the number of constraints $C \in \mathcal{I}$ for which $v \in \text{Var}(C)$. Moreover, we let the *degree* of $\mathcal{I}$ be the maximum degree of any variable $v \in \text{Var}(\mathcal{I})$.

An assignment $\alpha : \text{Var}(\mathcal{I}) \to D$ is a *solution* for a finite set $\mathcal{I}$ of constraints if it simultaneously satisfies all the constraints in $\mathcal{I}$. A finite set $\mathcal{I}$ of constraints is *satisfiable* if there exists a solution for it. The *Constraint Satisfaction Problem* (CSP, for short) asks, given a finite set $\mathcal{I}$ of constraints, whether $\mathcal{I}$ is satisfiable. By CSP(Γ) we denote the CSP restricted to instances $\mathcal{I}$ with $\text{Rel}(\mathcal{I}) \subseteq \Gamma$. A constraint language is *tractable* if for every finite subset $\Gamma' \subseteq \Gamma$, the problem CSP(Γ) can be solved in polynomial time.

We call a constraint language Γ *Boolean* if its underlying domain is $\{0, 1\}$, i.e., if it is a subset of $\mathbf{R}_{\{0,1\}}$. We consider the following properties of Boolean constraint languages. Let R be an n-ary relation. We say that R is:

— *0-valid* if $(0, \dots, 0) \in R$;
— *1-valid* if $(1, \dots, 1) \in R$;
— *Horn* if R is equivalent to a CNF formula that is Horn;
— *definite Horn* if R is equivalent to a CNF formula that is definite Horn;
— *anti-Horn* if R is equivalent to a CNF formula that is anti-Horn;
— *bijunctive* if R is equivalent to a CNF formula that is Krom;
— *affine* if R is equivalent to an affine formula; and
— *2-affine* if R is equivalent to a 2-affine formula.

We say that a constraint language Γ is 0-valid, 1-valid, Horn, definite Horn, anti-Horn, bijunctive, affine or 2-affine, respectively, if all relations $R \in \Gamma$ have this property.

In his seminal paper, Schaefer [1978] showed that for all constraint languages Γ over the Boolean domain $\{0, 1\}$, the CSP restricted to Γ is either NP-complete or solvable in polynomial time. In fact, he showed that a Boolean constraint language Γ is tractable if and only if it is 0-valid, 1-valid, Horn, anti-Horn, bijunctive or affine. A Boolean language that satisfies any of these six properties is called a *Schaefer language*.

We define what it means for a constraint language to be closed under partial assignment. Let $\alpha : X \to \mathcal{D}$ be an assignment. For an n-ary constraint $C = (S, R)$ with $S = (x_1, \dots, x_n)$ we denote by $C|_\alpha$ the constraint (S', R') obtained from C as follows. R' is obtained from R by (i) deleting all tuples $(d_1, \dots, d_n)$ from R for which there is some $1 \leq i \leq n$ with $\alpha(x_i) \neq d_i$, and removing from all remaining tuples all coordinates d_i with $x_i \in X$. S' is obtained from S by deleting all variables x_i with $x_i \in X$. For a set $\mathcal{I}$ of constraints we define $\mathcal{I}|_\alpha$ as $\{\, C|_\alpha : C \in \mathcal{I} \,\}$. We say that a constraint language Γ is *closed under partial assignment* if for any constraint C over Γ and any assignment $\alpha : X \to \mathcal{D}$ it holds that $C|_\alpha$ is also a constraint over Γ.

The parameterized problem of deciding whether a given CSP instance has an unsatisfiable subset of size k we denote by SMALL-CSP-UNSAT-SUBSET:

> SMALL-CSP-UNSAT-SUBSET
> *Instance:* A CSP instance $\mathcal{I}$, and a positive integer $k \geq 1$.
> *Parameter:* k.
> *Question:* Is there an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size k?

We also consider the following variant of SMALL-CSP-UNSAT-SUBSET, where the parameter additionally includes the maximum arity of the constraints in the CSP instance.

> SMALL-CSP-UNSAT-SUBSET[arity]
> *Instance:* A CSP instance $\mathcal{I}$, and a positive integer $k \geq 1$.
> *Parameter:* $k + a$, where a is the maximum arity of any constraint in $\mathcal{I}$.
> *Question:* Is there an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size k?

3. GENERAL CASE

The problem of deciding whether a set of constraints contains a small unsatisfiable subset has already been investigated from a parameterized complexity perspective for the case of CNF formulas by Fellows et al. [2006]. They proved that the problem SMALL-CNF-UNSAT-SUBSET is W[1]-complete in general. Moreover, their proof shows that hardness holds already for the case where each clause is of size at most 3. Since 3CNF formulas can be transformed to equivalent Boolean CSP instances in polynomial time, this implies that the problem SMALL-CSP-UNSAT-SUBSET is W[1]-hard, even when restricted to Boolean domains.

PROPOSITION 3.1 ([FELLOWS ET AL. 2006]). *SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET are W[1]-hard, even when restricted to ternary constraints on a Boolean domain.*

3.1. Hardness for W[2]

In the general case, we can strengthen this W[1]-hardness result for SMALL-CSP-UNSAT-SUBSET. The following result shows that the problem of identifying a small unsatisfiable subset of a CSP instance is W[2]-hard already when restricted to unary constraints.

PROPOSITION 3.2. SMALL-CSP-UNSAT-SUBSET *restricted to CSP instances with maximum arity* 1 *is* W[2]*-hard.*

PROOF. We give an fpt-reduction from HITTING-SET. Let $(U, \mathcal{T}, k)$ be an instance of HITTING-SET, where $\mathcal{T} = \{T_1, \dots, T_m\}$ is a family of subsets of the universe $U = \{u_1, \dots, u_n\}$. The idea behind this reduction is the following. We introduce a single variable v, and we introduce one domain element d_j for each subset $T_j \in \mathcal{T}$. Moreover, we introduce a single constraint C_i for each element u_i in the universe U. Intuitively, the constraint C_i rules out that the variable v is assigned values d_j corresponding to subsets T_j that are hit by the element u. Then, any set of constraints that rules out all assignments of the variable corresponds to a hitting set.

Formally, we construct a CSP instance $\mathcal{I}$ over a domain D as follows. We let Var$(\mathcal{I}) = \{v\}$ consist of a single variable, and we let its domain $D = \{d_1, \dots, d_m\}$ consist of one value d_j for each set T_j. We construct the set $\mathcal{I}$ of constraints as follows. The scope of all constraints contains only the variable v. Next, for each element $u_i \in U$ we introduce a constraint C_i that ensures that $v \in \{\, d_j : 1 \leq j \leq m, u_i \notin T_j \,\}$. This constraint C_i rules out the values d_j corresponding to the sets T_j that contain u_i. Ruling out all sets T_j with k of these constraints then corresponds exactly to finding a hitting set of size k. We claim that $(\mathcal{I}, k) \in$ SMALL-CSP-UNSAT-SUBSET if and only if $(U, \mathcal{T}, k) \in$ HITTING-SET.

($\Rightarrow$) Assume that there exists an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ that contains k constraints. Then there is some $\ell \leq k$ and some $1 \leq i_1 < \dots < i_\ell \leq n$ such that $C_{i_j} \in \mathcal{I}'$ for all $1 \leq j \leq \ell$. We claim that $U' = \{u_{i_1}, \dots, u_{i_\ell}\}$ is a hitting set of $\mathcal{T}$ of size $\ell \leq k$. We proceed indirectly, and assume that there is a set $T_j \in \mathcal{T}$ such that $U' \cap T_j = \emptyset$. Then the assignment α with $\alpha(v) = d_j$ is a solution for I', which is a contradiction.

($\Leftarrow$) Conversely, assume that there exists some $\ell \leq k$ and some $1 \leq i_1 < \dots < i_\ell \leq n$ such that $U' = \{u_{i_1}, \dots, u_{i_\ell}\}$ is a hitting set of $\mathcal{T}$. We claim that the subset $\mathcal{I}' \subseteq \mathcal{I}$ with $\mathcal{I}' = \{\, C_{i_j} : 1 \leq j \leq \ell \,\}$, containing at most k constraints, is unsatisfiable. We proceed indirectly and assume that there is a solution α for I'. Then, $\alpha(v) = d_j$ for some $1 \leq j \leq m$. From this we can conclude that $T_j \cap U' = \emptyset$, which contradicts the assumption that U' is a hitting set of $\mathcal{T}$. $\square$

Moreover, hardness for the class W[2] even holds when the domain is restricted to be Boolean.

PROPOSITION 3.3. *Given a CSP instance $\mathcal{I} = \{C_1, \dots, C_m\}$, in polynomial time we can construct a Boolean CSP instance $\mathcal{I}' = \{C'_1, \dots, C'_m\}$ such that each subset $\mathcal{I}_s \subseteq \mathcal{I}$ is satisfiable if and only if the corresponding subset $\mathcal{I}'_s = \{\, C'_i : C_i \in \mathcal{I}_s \,\}$ is satisfiable.*

PROOF. Suppose that the domain D of $\mathcal{I}$ is non-Boolean, and let Var$(\mathcal{I}) = \{x_1, \dots, x_n\}$. We let the Boolean CSP instance $\mathcal{I}'$ contain the variables Var$(\mathcal{I}') = \{\, x_{i,d} : 1 \leq i \leq n, d \in D \,\}$. Intuitively, a variable $x_{i,d}$ will represent whether x_i is assigned to value d.

We now specify how to construct the constraints C'_j. Take an arbitrary C_j, for some $1 \leq j \leq m$. We let Var$(C'_j) = \{\, x_{i,d} : x_i \in \text{Var}(C_j), d \in D \,\}$. Moreover, we construct the constraint relation of C'_j as follows. For each tuple $\overline{r}$ in the constraint relation of C_j, we add a tuple $\overline{r}'$ to the constraint relation of C'_j. For each variable $x_{i,d} \in \text{Var}(C'_j)$, the tuple $\overline{r}'$ sets $x_{i,d}$ to 1 if the tuple $\overline{r}$ sets x_i to d, and it sets $x_{i,d}$ to 0 otherwise.

Any assignment $\alpha : \text{Var}(\mathcal{I}) \to D$ then naturally corresponds to the assignment $\alpha' : \text{Var}(\mathcal{I}') \to \{0, 1\}$ that sets a variable $x_{i,d}$ to 1 if and only if α sets x_i to d. Using this correspondence, it is straightforward to verify that any subset $\mathcal{I}_s \subseteq \mathcal{I}$ is satisfiable if and only if the corresponding subset $\mathcal{I}'_s \subseteq \mathcal{I}'$ is satisfiable. $\square$

COROLLARY 3.4. SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances is* W[2]*-hard.*

PROOF. The fpt-reduction used in the proof of Proposition 3.3 is an fpt-reduction from SMALL-CSP-UNSAT-SUBSET to the problem SMALL-CSP-UNSAT-SUBSET restricted to Boolean CSP instances. By Proposition 3.2, we know that the former problem is W[2] hard. Therefore, we also know that SMALL-CSP-UNSAT-SUBSET restricted to Boolean CSP instances is W[2]-hard. □

3.2. Hardness for co-W[1]

A natural question to ask is whether the above W[2]-hardness results are tight, that is, whether they can be extended to W[2]-completeness results. We provide evidence that, even for Boolean instances, this is not the case. Concretely, we show that SMALL-CSP-UNSAT-SUBSET restricted to Boolean CSP instances is co-W[1]-hard. In order to do so, we consider the following parameterized problem.

SMALL-CSP-UNSAT
Instance: A CSP instance $\mathcal{I}$.
Parameter: The number k of constraints in $\mathcal{I}$.
Question: Is $\mathcal{I}$ unsatisfiable?

We show that SMALL-CSP-UNSAT is co-W[1]-complete, and co-W[1]-hard even when restricted to Boolean CSP instances. This has the consequence that SMALL-CSP-UNSAT-SUBSET is also co-W[1]-hard, even when restricted to Boolean CSP instances, because the identity mapping can be used to construct an fpt-reduction from SMALL-CSP-UNSAT to SMALL-CSP-UNSAT-SUBSET. Therefore, SMALL-CSP-UNSAT-SUBSET is not in W[2], unless W[2] $\subseteq$ co-W[1], which would imply that W[1] = co-W[1].

PROPOSITION 3.5. SMALL-CSP-UNSAT *is* co-W[1]-*complete. Moreover,* co-W[1]-*hardness holds even when the problem is restricted to Boolean CSP instances.*

PROOF. To show co-W[1]-hardness, we give an fpt-reduction from co-MULTI-COLORED-CLIQUE to SMALL-CSP-UNSAT. Let (V, E, k) be an instance of MULTI-COLORED-CLIQUE, where V is partitioned into the sets $V_1, \dots, V_k$. We assume without loss of generality that $|V_i| = n$, for each $1 \leq i \leq k$. Let $V_i = \{v_1^i, \dots, v_n^i\}$, for each $1 \leq i \leq k$. We construct a CSP instance $\mathcal{I}$ over the domain $D = \{0, 1\}$ with $k' = \binom{k}{2}$ constraints, that is satisfiable if and only if there is a k-clique in (V, E) that contains one vertex in each V_i.

We let Var$(\mathcal{I}) = \{\, x_j^i : 1 \leq i \leq k, 1 \leq j \leq n \,\}$. Then, we introduce a constraint $C_{i,j}$ for each $1 \leq i < j \leq k$. For each such constraint $C_{i,j} = (S_{i,j}, R_{i,j})$, we let:

$$S_{i,j} = (x_1^i, \dots, x_n^i, x_1^j, \dots, x_n^j).$$

We let the constraint relation $R_{i,j}$ consist of all (binary) tuples that (1) assign exactly one $x_{\ell_1}^i$ to 1, that (2) assign exactly one $x_{\ell_2}^j$ to 1, and for which (3) $\{v_{\ell_1}^i, v_{\ell_2}^j\} \in E$. Note that the number of tuples in $R_{i,j}$ is upper bounded by n^2.

We now show that $\mathcal{I} \notin$ SMALL-CSP-UNSAT if and only if $(V, E, k) \in$ MULTI-COLORED-CLIQUE.

($\Rightarrow$) Assume that there is an assignment α : Var$(\mathcal{I}) \to \{0, 1\}$ that satisfies $\mathcal{I}$. Then by construction of the constraints $C_{i,j}$, we know that for each $1 \leq i \leq k$, α assigns exactly one x_ℓ^i to 1. Consider the set $V' = \{\, v_{\ell_i}^i : 1 \leq i \leq k, \alpha(x_{\ell_i}^i) = 1 \,\}$. By definition, $|V' \cap V_i| = 1$ for each $1 \leq i \leq k$, and thus $|V'| = k$. We show that V' is a clique. Take arbitrary $1 \leq i < j \leq k$. Because α satisfies constraint $C_{i,j}$, we know that $\{v_{\ell_i}^i, v_{\ell_j}^j\} \in E$. Since i and j were arbitrary, we can conclude that V' is a clique, and thus that $(V, E, k) \in$ MULTI-COLORED-CLIQUE.

($\Leftarrow$) Conversely, suppose that there is a clique $V' \subseteq V$ such that $|V' \cap V_i| = 1$ for each $1 \leq i \leq k$. Consider the assignment α : Var$(\mathcal{I}) \to \{0, 1\}$, where $\alpha(x_\ell^i) = 1$ if and only if $v_\ell^i \in V'$, for each $1 \leq i \leq k$ and each $1 \leq \ell \leq n$. It is readily verified that α satisfies each constraint $C_{i,j}$. Thus, $\mathcal{I} \notin$ SMALL-CSP-UNSAT.

Next, to show membership in co-W[1], we give an fpt-reduction from SMALL-CSP-UNSAT to CLIQUE. Let $\mathcal{I}$ be an instance of SMALL-CSP-UNSAT, where $\mathcal{I} = \{C_1, \dots, C_k\}$. Moreover, for each $1 \leq i \leq k$, let $C_i = (S_i, R_i)$ and let $R_i = \{\overline{r}^i_1, \dots, \overline{r}^i_{\ell_i}\}$ consist of ℓ_i tuples $\overline{r}^i_j$. We construct an instance (V, E, k) of CLIQUE such that (V, E) contains a k-clique if and only if $\mathcal{I}$ is satisfiable.

We let the set $V = \{\, v^i_j : 1 \leq i \leq k, 1 \leq j \leq \ell_i \,\}$ of vertices contain a vertex for each tuple $\overline{r}^i_j$ in the constraint relation R_i of each constraint C_i. Then, we construct the edge set E as follows. For each two vertices v^i_j and $v^{i'}_{j'}$, we let $\{v^i_j, v^{i'}_{j'}\} \in E$ if and only if (1) $i \neq i'$, and (2) the tuples $\overline{r}^i_j$ and $\overline{r}^{i'}_{j'}$ are not conflicting, i.e., it is not the case that there is a variable $x \in \text{Var}(\mathcal{I})$ such that the truth assignments $\alpha^i_j : \text{Var}(C_i) \rightarrow D$ and $\alpha^{i'}_{j'} : \text{Var}(C_{i'}) \rightarrow D$ corresponding to $\overline{r}^i_j$ and $\overline{r}^{i'}_{j'}$, respectively, assign different values to x.

We show that $\mathcal{I} \notin$ SMALL-CSP-UNSAT if and only if $(V, E, k) \in$ CLIQUE.

($\Rightarrow$) Assume that there is some assignment $\alpha : \text{Var}(\mathcal{I}) \rightarrow D$ that satisfies $\mathcal{I}$. Consider the set $V' \subset V$ defined as follows. For each $1 \leq i \leq k$ and each $1 \leq j \leq \ell_i$, the set V' contains the unique vertex v^i_j such that (the assignment corresponding to) the tuple $\overline{r}^i_j$ agrees with α. Since for each i there is a unique tuple r^i_j that agrees with α, we get that $|V'| = k$.

We show that V' is a clique. Take two arbitrary vertices $v^i_j, v^{i'}_{j'} \in V'$. By definition of V', the tuples $\overline{r}^i_j$ and $\overline{r}^{i'}_{j'}$ both agree with α. Therefore, they cannot be conflicting. Then, by definition of E, we get that $\{v^i_j, v^{i'}_{j'}\} \in E$. Since v^i_j and $v^{i'}_{j'}$ were arbitrary, we can conclude that V' is a clique. Thus, $(V, E, k) \in$ CLIQUE.

($\Leftarrow$) Conversely, suppose that there is a k-clique $V' \subseteq V$ of (V, E). We show that $\mathcal{I}$ is satisfiable. By construction of E, no two vertices v^i_j and $v^{i'}_{j'}$ with $i = i'$ are connected by an edge. Therefore, V' contains exactly one vertex $v^i_{j_i}$ for each $1 \leq i \leq k$. We show that the assignments corresponding to the tuples $\overline{r}^i_{j_i}$ can be combined into one single assignment $\alpha : \text{Var}(\mathcal{I}) \rightarrow D$ that satisfies $\mathcal{I}$. All we have to show is that for each two $1 \leq i < i' \leq k$, the tuples $\overline{r}^i_{j_i}$ and $\overline{r}^{i'}_{j_{i'}}$ are not conflicting. By our assumption that $v^i_{j_i}$ and $v^{i'}_{j_{i'}}$ are connected by an edge, and by construction of E, this is the case. Therefore, $\mathcal{I} \notin$ SMALL-CSP-UNSAT. $\square$

COROLLARY 3.6. SMALL-CSP-UNSAT-SUBSET *is not in* W[2]*, unless* W[1] = co-W[1].

3.3. Completeness for A[2]

In this section, we show that the problem SMALL-CSP-UNSAT-SUBSET, in its unrestricted form, is A[2]-complete. Moreover, we show that hardness for A[2] holds even when we restrict the problem to Boolean CSP instances.

For the case where the arity of the constraints is bounded, the strongest intractability result for SMALL-CSP-UNSAT-SUBSET that we establish remains the W[2]-hardness result from Section 3.1 (Proposition 3.2). The A[2]-hardness proof that we give in this section does not work for the case where the arity of constraints is bounded. It remains open to establish a completeness result for the problem SMALL-CSP-UNSAT-SUBSET restricted to constraints of bounded arity—interestingly, the co-W[1]-hardness result from Section 3.2 suggests that the problem is not complete for W[2].

We now turn our attention to establishing the A[2]-completeness of SMALL-CSP-UNSAT-SUBSET. First, we show that the problem SMALL-CSP-UNSAT-SUBSET is A[2]-hard, even when restricted to Boolean CSP instances. Second, we show its membership in A[2].

PROPOSITION 3.7. SMALL-CSP-UNSAT-SUBSET *is* A[2]*-hard.*

PROOF. We show A[2]-hardness by means of an fpt-reduction from MC(Σ_2). Take an arbitrary instance of MC(Σ_2), consisting of a structure $\mathcal{A}$ with universe A (over the fixed binary signature τ), and a first-order formula $\varphi = \exists x_1, \dots, x_k \forall y_1, \dots, y_k\ \psi$, where ψ is a disjunction of atoms. We

Technical Report AC-TR-17-009

construct a CSP instance $\mathcal{I}$ and an integer k', such that $(\mathcal{I}, k') \in$ SMALL-CSP-UNSAT-SUBSET if and only if $\mathcal{A} \models \varphi$.

We define $k' = \binom{k}{2} + k + 1$. We let the domain of $\mathcal{I}$ be $D = A \cup \{0, 1, \dots, k\}$. We may assume without loss of generality that $A \cap \{0, 1, \dots, k\} = \emptyset$. We introduce the following variables:

$$\begin{aligned} \text{Var}(\mathcal{I}) &= V \cup W \cup X \cup Z, \text{ where} \\ V &= \{v\}, \\ W &= \{\, w_i : 1 \leq i \leq k \,\}, \\ X &= \{\, x_{i,a} : 1 \leq i \leq k, a \in A \,\}, \\ Y &= \{\, y_i : 1 \leq i \leq k \,\}, \text{ and} \\ Z &= \{\, z_{i,j} : 1 \leq i < j \leq k \,\}. \end{aligned}$$

The constraints in $\mathcal{I}$ are defined as follows. For each $1 \leq i < j \leq k$ and each $a_1, a_2 \in A$, we introduce a constraint C_{i,j,a_1,a_2} for which $\text{Var}(C_{i,j,a_1,a_2}) = \{v, x_{i,a_1}, x_{j,a_2}, y_i, y_j, z_{i,j}\}$. The constraint relation has the following $k^2|A|^2 + 1$ tuples. For each $0 \leq i' \leq k$ such that $i \neq i'$, for each $0 \leq j' \leq k$ such that $j \neq j'$, and for each $a_3, a_4 \in A$, there is a tuple $\overline{r}_{i',j',a_3,a_4}$. Each such tuple $\overline{r}_{i',j',a_3,a_4}$ sets x_{i,a_1} to j', sets x_{j,a_2} to i', sets y_i to a_3, sets y_j to a_4, and sets v to 1. Moreover, it sets $z_{i,j}$ to 1 if at least one atom in ψ containing only variables among x_i, x_j, y_i, y_j is satisfied by the partial assignment $\alpha = \{x_i \mapsto a_1, x_j \mapsto a_2, y_i \mapsto a_3, y_j \mapsto a_4\}$; otherwise it sets $z_{i,j}$ to 0. In addition, there is a tuple that sets all variables in $\text{Var}(C_{i,j,a_1,a_2})$ to 0. In particular, this constraint rules out that x_{i,a_1} is set to j and that x_{j,a_2} is set to i.

Then, for each $1 \leq i \leq k$ and each $a \in A$, we introduce a constraint $C_{i,a}$ for which $\text{Var}(C_{i,a}) = \{x_{i,a}, w_i\}$. The constraint relation has $2k + 1$ tuples. For each $1 \leq j \leq k$ and each $0 \leq b \leq 1$, there is a tuple that sets $x_{i,a}$ to j and w_i to b. In addition, there is a tuple that sets $x_{i,a}$ to 0 and that sets w_i to 1. In other words, this constraint enforces that w_i is set to 1 if $x_{i,a}$ cannot be set to any value $j > 0$.

Finally, we introduce a constraint C_0 for which $\text{Var}(C_0) = V \cup W \cup Z$. The constraint relation has $(2^k - 1) \cdot 2^{k''} + 1$ tuples, where $k'' = \binom{k}{2}$. For each assignment $\rho : W \cup Z \rightarrow \{0, 1\}$ such that at for at least one $1 \leq i \leq k$ it holds that $\rho(w_i) = 1$, there is a tuple that sets all variables according to ρ, and that sets v to 0. Moreover, there is an additional tuple that sets all variables $w_i \in W$ to 1, that sets all variables $z_{i,j} \in Z$ to 0, and that sets v to 1. In other words, C_0 is satisfied if and only if either (1) at least one w_i is set to 0, or (2) all w_i are set to 1 and all $z_{i,j}$ are set to 1.

Before we show the correctness of this reduction, we argue that each unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ must include C_0. Suppose that this is not the case. Then the assignment that sets all variables $w_i \in W$ to 1 and that sets all other variables to 0 satisfies $\mathcal{I}'$.

Also, we argue that each unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ must include some constraint C_{i,a_i} for each $1 \leq i \leq k$, and some $a_i \in A$. Suppose that this is not the case. Moreover, let $1 \leq i_1 < \dots < i_m \leq k$ be the indices for which $\mathcal{I}'$ does contain some constraint $C_{i_\ell,a_{i_\ell}}$. We know that there is some $1 \leq i_0 \leq k$ such that $\mathcal{I}'$ contains no constraint $C_{i_0,a}$ (for $a \in A$). Then consider the assignment $\alpha : \text{Var}(\mathcal{I}) \rightarrow D$ that sets w_{i_ℓ} to 1 for each $1 \leq \ell \leq m$, and that sets all other variables to 0. Then α satisfies all constraints C_{i,j,a_1,a_2} and $C_{i,a}$ in $\mathcal{I}'$. Since α sets at least one variable w_i to 1, we also know that α satisfies C_0. Therefore, α must satisfy all constraints in $\mathcal{I}'$, which is a contradiction with our assumption that $\mathcal{I}'$ is unsatisfiable.

Finally, we argue that for each unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size k' there must be some assignment $\alpha : \{1, \dots, k\} \rightarrow A$ such that $\mathcal{I}'$ contains the constraints $C_{i,\alpha(i)}$ for each $1 \leq i \leq k$, and the constraints $C_{i,j,\alpha(i),\alpha(j)}$ for each $1 \leq i < j \leq k$. Suppose that this is not the case. By a straightforward counting argument, we then know that it must be the case that for some $1 \leq i_0 \leq k$, and some $1 \leq j_0 \leq k$ such that $i_0 \neq j_0$, there is some $C_{i_0,a_{i_0}} \in \mathcal{I}'$ but for no a_{j_0} the constraint $C_{i_0,j_0,a_{i_0},a_{j_0}}$ is in $\mathcal{I}'$. (We implicitly assume that $j_0 > i_0$; the case where $j_0 < i_0$ is entirely analogous.) Then consider the assignment $\alpha : \text{Var}(\mathcal{I}) \rightarrow D$ that sets $x_{i_0,a_{i_0}}$ to j_0, that sets w_{i_0} to 0, and sets all other variables in Var($\mathcal{I}$) to 0. The only constraints that are conflicting with $\alpha(x_{i_0,a_{i_0}}) = j_0$ are constraints $C_{i_0,j_0,a_{i_0},a_{j_0}}$, which by assumption are not contained in $\mathcal{I}'$. Therefore, it is readily verified that α satisfies all constraints in $\mathcal{I}'$. This is a contradiction with our assumption that $\mathcal{I}'$ is unsatisfiable.

We are now ready to prove that $\mathcal{I}$ has an unsatisfiable subset $\mathcal{I}'$ of size k' if and only if $\mathcal{A} \models \varphi$, i.e., that $(\mathcal{I}, k') \in$ SMALL-CSP-UNSAT-SUBSET if and only if $(\mathcal{A}, \varphi) \in$ MC(Σ_2).

($\Rightarrow$) Suppose that there is an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of k' constraints. As we showed above, then $C_0 \in \mathcal{I}'$. Moreover there exists some assignment $\alpha : \{1, \dots, k\} \to A$ such that $\mathcal{I}'$ contains the constraints $C_{i,\alpha(i)}$ for each $1 \leq i \leq k$, and the constraints $C_{i,j,\alpha(i),\alpha(j)}$ for each $1 \leq i < j \leq k$. Now consider the assignment $\alpha' : X \to A$ defined by letting $\alpha'(x_i) = \alpha(i)$, for each $1 \leq i \leq k$. We claim that $\mathcal{A}, \alpha' \models \forall y_1, \dots, y_k\ \psi$. Take an arbitrary assignment $\beta' : Y \to A$. We show that $\mathcal{A}, \alpha' \cup \beta' \models \psi$. Consider the assignment β : Var($\mathcal{I}$) $\to D$ defined by letting $\beta(y_i) = \beta'(y_i)$ for all $y_i \in Y$, letting $\beta(x_{i,a}) = 0$ for all $x_{i,a} \in X$, letting $\beta(v) = 1$, letting $\beta(z_{i,j}) = 0$ for all $z_{i,j} \in Z$, and letting $\beta(w_i) = 1$ for all $w_i \in W$. Since $\mathcal{I}'$ is unsatisfiable, we know that β cannot satisfy all constraints in $\mathcal{I}'$. The only possible contradiction is that some constraint $C_{i_0,j_0,\alpha(i_0),\alpha(j_0)} \in \mathcal{I}'$ does not allow z_{i_0,j_0} to be set to 0. By construction of $C_{i_0,j_0,\alpha(i_0),\alpha(j_0)}$, this is only the case if $\alpha' \cup \beta'$ satisfies some atom in ψ. Therefore, we can conclude that $\mathcal{A}, \alpha' \cup \beta' \models \psi$. Since β' was arbitrary, we know that $\mathcal{A}, \alpha' \models \forall y_1, \dots, y_k\ \psi$. Therefore, $\mathcal{A} \models \varphi$, and thus $(\mathcal{A}, \varphi) \in$ MC(Σ_2).

($\Leftarrow$) Conversely, suppose that $\mathcal{A} \models \varphi$, that is, that there exists some assignment α : $\{x_1, \dots, x_k\} \to A$ such that $\mathcal{A}, \alpha \models \forall y_1, \dots, y_k\ \psi$. Consider the following subset $\mathcal{I}' \subseteq \mathcal{I}$ of constraints:

$$\mathcal{I}' = \{C_0\} \cup \{\, C_{i,\alpha(x_i)} : 1 \leq i \leq k \,\} \cup \{\, C_{i,j,\alpha(x_i),\alpha(x_j)} : 1 \leq i < j \leq k \,\}.$$

Clearly, $|\mathcal{I}'| = k'$. We claim that $\mathcal{I}'$ is unsatisfiable. To derive a contradiction, suppose that $\mathcal{I}'$ is satisfiable, i.e., that there is an assignment β' : Var($\mathcal{I}$) $\to D$ that satisfies all constraints in $\mathcal{I}'$. Take an arbitrary $1 \leq i \leq k$. Since $\mathcal{I}'$ contains the constraints $C_{i,j,\alpha(x_i),\alpha(x_j)}$ for all $j > i$, and the constraints $C_{j,i,\alpha(x_j),\alpha(x_i)}$ for all $j < i$, we know that β' must set $x_{i,\alpha(x_i)}$ to 0, and thus since $\mathcal{I}'$ contains the constraint $C_{i,\alpha(x_i)}$, we know that β' must set w_i to 1. Then, since $\mathcal{I}'$ contains C_0, we know that β' sets all $z_{i,j} \in Z$ to 0, and that β' sets v to 1. Moreover, since $\mathcal{I}'$ contains the constraints $C_{i,j,\alpha(x_i),\alpha(x_j)}$ for all $1 \leq i < j \leq k$, we know that $\beta'(y_i) \in A$, for all $y_i \in Y$. Then, consider the restriction $\beta : \{y_1, \dots, y_k\} \to A$ of β' to the variables in Y, i.e., $\beta(y_i) = \beta'(y_i)$ for all $y_i \in Y$. Since $\mathcal{A}, \alpha \models \forall y_1, \dots, y_k\ \psi$, we know that $\mathcal{A}, \alpha \cup \beta \models \psi$, that is, there must be some atom R in ψ that is satisfied by $\alpha \cup \beta$. Let $1 \leq i_0 < j_0 \leq k$ be indices such that the variables in R are among $\{x_{i_0}, x_{j_0}, y_{i_0}, y_{j_0}\}$ (we know such i_0, j_0 exist, because R is unary or binary). Then by construction of $C_{i_0,j_0,\alpha(x_{i_0}),\alpha(x_{j_0})}$, we know that β' is forced to set z_{i_0,j_0} to 1. This is a contradiction with our previous conclusion that $\beta'(z_{i,j}) = 0$ for all $z_{i,j} \in Z$. Therefore, we can conclude that $\mathcal{I}'$ is unsatisfiable, and that $(\mathcal{I}, k') \in$ SMALL-CSP-UNSAT-SUBSET. $\square$

COROLLARY 3.8. SMALL-CSP-UNSAT-SUBSET *is* A[2]*-hard, even when restricted to Boolean CSP instances.*

PROOF. Proposition 3.3 directly gives us an fpt-reduction to the problem SMALL-CSP-UNSAT-SUBSET restricted to Boolean CSP instances. $\square$

PROPOSITION 3.9. SMALL-CSP-UNSAT-SUBSET *is in* A[2].

PROOF. We show membership in A[2] by fpt-reducing the problem to MC(Σ_2). Let $(\mathcal{I}, k)$ be an instance of SMALL-CSP-UNSAT-SUBSET, with $\mathcal{I} = \{C_1, \dots, C_m\}$, and $C_i = (S_i, R_i)$ for each $1 \leq i \leq m$. Moreover, let u be the maximum number of tuples in any of the constraint relations R_i.

We construct an instance $(\mathcal{A}, \varphi)$ of $\text{MC}(\Sigma_2)$ (over a fixed signature τ) as follows. We define the universe A of $\mathcal{A}$ as follows:

$$A = \{c_1, \dots, c_m\} \cup \{t_1, \dots, t_u\}.$$

Intuitively, the elements c_i represent the constraints C_i, and the elements t_j represent tuples in constraint relations R_i.

We introduce unary relations C and T to τ. Intuitively, C encodes whether an element represents a constraint, and T encodes whether an element represents a tuple. We let:

$$C^{\mathcal{A}} = \{c_1, \dots, c_m\} \qquad \text{and} \qquad T^{\mathcal{A}} = \{t_1, \dots, t_u\}.$$

We introduce a binary relation I to τ that intuitively encodes whether a constraint C_i has at least j tuples. We let:

$$I^{\mathcal{A}} = \{\, (c_i, t_j) : |R_i| \geq j \,\}.$$

Finally, we introduce a 4-ary relation N to τ, that intuitively represents whether a tuple t_{j_1} in the constraint relation R_{i_1} and a tuple t_{j_2} in the constraint relation R_{i_2} are non-conflicting. (Moreover, it ensures that there is a j_1-th tuple in R_{i_1} and a j_2-th tuple in R_{i_2}.) We let:

$$\begin{aligned} N^{\mathcal{A}} = \{\, & (c_{i_1}, t_{j_1}, c_{i_2}, t_{j_2}) : 1 \leq i_1 \leq m, 1 \leq j_1 \leq |R_{i_1}|, 1 \leq i_2 \leq m, 1 \leq j_2 \leq |R_{i_2}|, \\ & \text{the } j_1\text{-th tuple in } R_{i_1} \text{ and the } j_2\text{-th tuple in } R_{i_2} \text{ are not conflicting} \,\}. \end{aligned}$$

Then, we define the first-order logic sentence φ as follows:

$$\begin{aligned} \varphi = {} & \exists x_1, \dots, x_k \forall y_1, \dots, y_k \\ & \bigwedge_{1 \leq i \leq k} C(x_i) \wedge \left(\bigwedge_{1 \leq i \leq k} (T(y_i) \wedge I(x_i, y_i)) \right) \rightarrow \left(\bigvee_{1 \leq i < j \leq k} \neg N(x_i, y_i, x_j, y_j) \right). \end{aligned}$$

Clearly $|\varphi| \leq f(k)$ for some function f.

Any assignment $\alpha : \{x_1, \dots, x_k\} \rightarrow \{c_1, \dots, c_m\}$ then naturally corresponds to a subset $\mathcal{I}' \subseteq \mathcal{I}$ of size k. Also, any subsequent assignment $\beta : \{y_1, \dots, y_k\} \rightarrow \{t_1, \dots, t_u\}$ that for each $1 \leq i \leq k$ assigns y_i to a sufficiently small value t_j (that is, $1 \leq j \leq |R_\ell|$ where $\alpha(x_i) = c_\ell$) naturally corresponds to a choice of a tuple t_j in the constraint relation R_i of each constraint $C_i \in \mathcal{I}'$. Using these correspondences, it is straightforward to verify that $(\mathcal{I}, k) \in$ SMALL-CSP-UNSAT-SUBSET if and only if $\mathcal{A} \models \varphi$. □

THEOREM 3.10. *Both* SMALL-CSP-UNSAT-SUBSET *and* SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances are* A[2]*-complete.*

3.3.1. Completeness for Σ_2^{p}. The A[2]-completeness result for SMALL-CSP-UNSAT-SUBSET suggests that the unparameterized variant of the problem is complete for Σ_2^{p}, as A[2] is a parameterized analogue of Σ_2^{p}. For the case of CNF formulas, this is known to be the case [Liberatore 2005]. We show that this Σ_2^{p}-completeness result can be extended to the case of CSP instances.

Formally, we consider the (non-parameterized) problem CSP-UNSAT-SUBSET, where instances consist of a CSP instance $\mathcal{I}$ and a positive integer $m \geq 1$. The question is whether there exists an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size m. The (non-parameterized) problem CNF-UNSAT-SUBSET is defined analogously for the case of CNF formulas. We use the following convention to denote the non-parameterized problems that we discuss in this section:

Remark 3.11. We indicate non-parameterized variants of the problem SMALL-UNSAT-SUBSET by a lack of the prefix SMALL- in the name of the problem.

For these problems, membership in Σ_2^{p} follows by an encoding in QSAT_2. For the case of CNF formulas, this has been shown by Liberatore [2005], and for the case of CSP instances, the known encoding can be extended straightforwardly. Hardness for Σ_2^{p} has also been shown by Liberatore [2005, Theorem 2], for the case of CNF formulas.

PROPOSITION 3.12 ([LIBERATORE 2005]). CNF-UNSAT-SUBSET *is* Σ_2^{p}*-complete.*

We show how to extend this result to the case of CSP instances. The most obvious way to show this would be to give a direct reduction from the problem CNF-UNSAT-SUBSET to CSP-UNSAT-SUBSET, by expressing each clause as a CSP constraint. However, if no bound on the clause size is known, this might result in an exponential blow-up in the size of the constraint relations needed to express clauses. Therefore, we show that the Σ_2^{p}-hardness of CNF-UNSAT-SUBSET even holds for the case where all clauses have bounded size.

PROPOSITION 3.13. CNF-UNSAT-SUBSET *is* Σ_2^{p}*-hard, even when restricted to CNF formulas containing only clauses of size at most* 3*.*

PROOF. Let (φ, m) be an instance of CNF-UNSAT-SUBSET. We show how to transform (φ, m) into an equivalent instance (φ_3, m_3) in polynomial time, where φ_3 contains only clauses of size at most 3.

Let u be the maximum size of any clause appearing in φ. We firstly transform φ into a formula φ_1 that contains an unsatisfiable subset of size $m_1 = m + 1$ if and only if φ contains an unsatisfiable subset of size m. Moreover, each unsatisfiable subset of φ_1 must contain one particular (unit) clause. Let $z \notin \mathrm{Var}(\varphi)$ be a fresh variable. We let $\varphi_1 = \{\{z\}\} \cup \{\, c \cup \{\overline{z}\} : c \in \varphi \,\}$. It is straightforward to show that each unsatisfiable subset of φ_1 must contain the clause $\{z\}$, and that there exists a natural correspondence between unsatisfiable subset of φ (of size m) and unsatisfiable subset of φ_1 (of size m_1).

Next, we transform φ_1 into a formula φ_2 that has only clauses that are either unit clauses, or clauses of size $u+1$. We introduce u fresh variables $z_1, \dots, z_u \notin \mathrm{Var}(\varphi_1)$. Then, for each clause $c \in \varphi_1$ of size $1 \leq \ell \leq u + 1$, we add the clause $c' = c \cup \{\overline{z_\ell}, \overline{z_{\ell+1}}, \dots, \overline{z_u}\}$ to φ_2. Moreover, we add the unit clauses $c_1, \dots, c_u$, where $c_i = \{z_i\}$ for each $1 \leq i \leq u$. to φ_2.

Firstly, we show that each unsatisfiable subset ψ of φ_2 must contain the unit clauses $c_1, \dots, c_u$. To derive a contradiction, suppose that this is not the case, that is, there is some $1 \leq i \leq u$ such that $c_i \notin \psi$. We know that ψ must contain the clause $\{z, \overline{z_1}, \dots, \overline{z_u}\}$. Moreover, we know that all other clauses in ψ contain the literal $\overline{z}$. Then any truth assignment that sets z to 0 and that sets z_i to 0 satisfies ψ, which is a contradiction with our assumption that ψ is unsatisfiable. Therefore, we can conclude that each unsatisfiable subset of φ_2 contains all of the clauses $c_1, \dots, c_u$.

Moreover, there exists a natural correspondence between unsatisfiable subsets of φ_1 of size m_1 and unsatisfiable subsets of φ_2 of size $m_2 = m_1 + u$. Therefore, (φ_1, m_1) and (φ_2, m_2) are equivalent instances of CNF-UNSAT-SUBSET. Moreover, we know that each unsatisfiable subset $\psi \subseteq \varphi_2$ of size m_2 contains exactly u unit clauses and m_1 clauses of size exactly $u + 1$.

We can now transform (φ_2, m_2) into an equivalent instance (φ_3, m_3) as follows. Each clause $c = \{l_1, \dots, l_{u+1}\} \in \varphi_2$ of size $u + 1$, we replace by the clauses $\{l_1, y_1^c\}$, $\{\overline{y_1^c}, l_2, y_2^c\}$, $\dots$, $\{\overline{y_{u-1}^c}, l_u, y_u^c\}$, and $\{\overline{y_u^c}, l_{u+1}\}$, where the variables $y_1^c, \dots, y_u^c$ are fresh variables. Clearly, φ_3 contains only clauses of size at most 3. Moreover, it is straightforward to verify that φ_2 contains an unsatisfiable subset of size m_2 if and only if φ_3 contains an unsatisfiable subset of size $m_3 = (u + 1)m_1 + u = (u + 1)(m + 1) + u$. Therefore, φ contains an unsatisfiable subset of size m if and only if φ_3 contains an unsatisfiable subset of size m_3. $\square$

This result allows us to give a polynomial-time reduction to the problem CSP-UNSAT-SUBSET, by expressing each clause of size 3 as a constraint over 3 variables and with a constraint relation containing 7 tuples.

COROLLARY 3.14. *The problem* CSP-UNSAT-SUBSET *is* Σ_2^{p}*-complete.*

3.4. W[1]-Membership for Small Arity and Domain Size

The fpt-reductions in the W[2]-hardness and A[2]-hardness proofs above all use CSP instances either with a large domain, or with constraints of large arity. We show that with bounds on both the domain size and the maximum arity of constraints, the problem SMALL-CSP-UNSAT-SUBSET is contained in W[1]. The general proof strategy will be the same as the W[1]-membership proof for SMALL-CNF-UNSAT-SUBSET by Fellows et al. [2006].

Their general proof strategy goes as follows. They use the result by Papadimitriou and Yannakakis [1999] that the model checking problem for first-order logic, parameterized by the size of the formula, is W[1]-complete for positive formulas. They then continue to show that for each parameter value k, the problem SMALL-CNF-UNSAT-SUBSET can be expressed as an instance of the first-order model checking problem, where the first-order formula is positive and depends only on the set of minimal unsatisfiable formulas (of size at most k) on variables $y_1, \dots, y_{k-1}$. They thus give an fpt-reduction from SMALL-CNF-UNSAT-SUBSET to the problem MC(positive), thereby showing W[1]-membership.

In this argument, they use the underlying assumption that, for CNF formulas, any minimal unsatisfiable subset of size at most k contains at most $k-1$ variables. This is known as Tarsi's Lemma [Aharoni and Linial 1986; Kullmann 2000b]. For the case of CSP instances, we will need a similar lemma that bounds the size and the number of possible sets of constraints whose arity and domain size depend only on the parameter value.

LEMMA 3.15. *Let $\mathcal{C}$ be a class of CSP constraints on the domain $D = \{d_1, \dots, d_{k_1}\}$, for some $k_1 \in \mathbb{N}$, where each constraint $C \in \mathcal{C}$ has arity at most $k_2 \in \mathbb{N}$. Moreover, let $k_0 \in \mathbb{N}$. Then, the number of unsatisfiable sets $\mathcal{I}$ of size at most k_0, containing only constraints in $\mathcal{C}$ and containing only the variables $x_1, \dots, x_{k_0 \cdot k_2}$, is bounded above by $f(k_0, k_1, k_2)$, for some computable function $f : \mathbb{N}^3 \to \mathbb{N}$. Moreover, these sets can be enumerated in time $g(k_0, k_1, k_2)$, for some computable function $g : \mathbb{N}^3 \to \mathbb{N}$.*

PROOF. Since each constraint in $\mathcal{C}$ contains at most k_2 variables, we know that any set of k_0 constraints contains at most $u = k_0 \cdot k_2$ variables. Take some unsatisfiable subset $\mathcal{I}$ of size at most k_0, containing only constraints in $\mathcal{C}$, and containing only the variables $x_1, \dots, x_u$. By a straightforward counting argument we get that the constraints in $\mathcal{C}$ contain at most μ_1 many different constraint relations, where:

$$\mu_1 = \sum_{\ell=1}^{k_2} k_1^{\ell}.$$

Then, using only the variables $x_1, \dots, x_{k_0 \cdot k_2}$, one can construct at most μ_2 different constraints, where $\mu_2 = (k_0 \cdot k_2)^{k_2} \cdot \mu_1$. Therefore, there are at most μ_3 many different sets $\mathcal{I}$ of size at most k_0, containing only constraints in $\mathcal{C}$ and containing only the variables $x_1, \dots, x_{k_0 \cdot k_2}$, where $\mu_3 = (\mu_2)^{k_0}$. Then μ_3 is also an upper bound on the number of such sets $\mathcal{I}$ that are unsatisfiable.

Enumerating all such sets $\mathcal{I}$ that are unsatisfiable can be done by simply going over all such possible sets $\mathcal{I}$, and for each set checking whether it is unsatisfiable. Since each $\mathcal{I}$ contains at most $k_0 \cdot k_2$ many variables, and has domain size k_1, this (un)satisfiability check can be done in time $O((k_0 \cdot k_2)^{k_1})$. □

Using this lemma, we can now prove W[1]-membership for SMALL-CSP-UNSAT-SUBSET when the arity and the domain size are bounded.

PROPOSITION 3.16. SMALL-CSP-UNSAT-SUBSET *is in* W[1] *when restricted to constraints of arity at most $f_1(k)$ over a domain of size at most $f_2(k)$, where k is the parameter value, and f_1, f_2 are computable functions.*

PROOF. We give an fpt-reduction to the problem MC(positive). Take an instance $(\mathcal{I}, k)$ of SMALL-CSP-UNSAT-SUBSET, where the arity of all constraints in $\mathcal{I}$ is at most $a = f_1(k)$, and where the domain size of $\mathcal{I}$ is at most $d = f_2(k)$. Let the domain of $\mathcal{I}$ be $D = \{d_1, \dots, d_{m_1}\}$, for some $m_1 \leq f_2(k)$. Also, let $\text{Var}(\mathcal{I}) = \{x_1, \dots, x_{u_1}\}$, and $\mathcal{I} = \{C_1, \dots, C_{u_2}\}$. We construct a positive first-order logic sentence φ and a relational structure $\mathcal{A}$ (with universe A) over the signature τ. The sentence φ depends only on k, and contains some constants $a \in A$.

By Lemma 3.15, we know that the number m_2 of possible unsatisfiable sets $\mathcal{I}'$ over the variables $x_1, \dots, x_{k \cdot a}$ and the domain D, containing only constraints of arity at most a, is at most $g_1(k)$, for some computable function g_1. Moreover, we can enumerate all such unsatisfiable sets $\mathcal{I}_1, \dots, \mathcal{I}_{m_2}$ in time $g_2(k)$, for some computable function g_2. (Without loss of generality, we only consider such sets $\mathcal{I}_i$ that are minimally unsatisfiable and that involve exactly the variables $x_1, \dots, x_b$ for some $1 \leq b \leq k \cdot a$.) Also, the number m_3 of different constraint relations that occur in the constraints in $\mathcal{I}$ is bounded by $g_3(k)$, for some computable function g_3. Moreover, all such constraint relations $R_1, \dots, R_{m_3}$ can be enumerated in time $g_4(k)$, for some computable function g_4.

We firstly construct the structure $\mathcal{A}$. We let the universe of $\mathcal{A}$ be $A = \{\, v_i : 1 \leq i \leq u_1 \,\} \cup \{1, \dots, m_3\} \cup \{\, c_i : 1 \leq i \leq u_2 \,\}$. Then, we introduce unary predicates V, C, N to τ, and we let $V^{\mathcal{A}} = \{\, v_i : 1 \leq i \leq u_1 \,\}$, $C^{\mathcal{A}} = \{\, c_i : 1 \leq i \leq u_2 \,\}$, and $N^{\mathcal{A}} = \{1, \dots, m_3\}$. Moreover, we introduce a binary predicate T to τ, and we let $T^{\mathcal{A}}$ consist of all tuples (c_i, ℓ), where $1 \leq i \leq u_2$, $1 \leq \ell \leq m_3$, and the constraint relation of $C_i \in \mathcal{I}$ is R_ℓ. Finally, we introduce a ternary predicate O to τ, and we let $O^{\mathcal{A}}$ consist of all tuples (c_i, j, v_ℓ), where $1 \leq i \leq u_2$, $1 \leq j \leq \text{arity}(C_i)$, $1 \leq \ell \leq u_1$, and the j-th variable in the scope of C_i is v_ℓ.

Next, we construct φ. For each such $\mathcal{I}_i$, for $1 \leq i \leq m_2$, we construct a positive first-order logic sentence φ_i, and we let:

$$\varphi = \bigvee_{1 \leq i \leq m_2} \varphi_i.$$

Take $1 \leq i \leq m_2$, and let $\mathcal{I}_i = \{C'_1, \dots, C'_{\mu_1}\}$, where $\mu_1 \leq k$. Moreover, let $\text{Var}(\mathcal{I}_i) = \{x_1, \dots, x_{\mu_2}\}$. Also, for each $1 \leq j \leq \mu_1$, let $C'_j = (S'_j, R'_j)$, where $S'_j = (x_{\nu_{j,1}}, \dots, x_{\nu_{j,b_j}})$, and where $R'_j = R_{\nu'_j}$. Then we define:

$$\varphi_i = \exists z_1, \dots, z_{\mu_2} \exists y_1, \dots, y_{\mu_1} \bigwedge_{1 \leq i \leq \mu_2} V(z_i) \wedge \bigwedge_{1 \leq j \leq \mu_1} \left(C(y_j) \wedge T(y_j, \nu'_j) \wedge \bigwedge_{1 \leq j' \leq b_j} O(y_j, j', z_{\nu_{j,j'}}) \right).$$

It is readily verified that $|\varphi|$ depends only on k, and that φ is a positive first-order logic sentence. We claim that $(\mathcal{I}, k) \in$ SMALL-CSP-UNSAT-SUBSET if and only if $\mathcal{A} \models \varphi$.

($\Rightarrow$) Take a minimally unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size $k' \leq k$. We know that there exists a bijection $\rho : \text{Var}(\mathcal{I}) \to \{x_1, \dots, x_{|\text{Var}(\mathcal{I})|}\}$ such that applying ρ to $\mathcal{I}'$ yields the set $\mathcal{I}_i$ for some $1 \leq i \leq m_2$. Let $\mathcal{I}_i = \{C'_1, \dots, C'_{k'}\}$ and let $\text{Var}(\mathcal{I}_i) = \{x_1, \dots, x_{\mu_2}\}$. Now, consider the assignment $\alpha : \{z_1, \dots, z_{\mu_2}\} \cup \{y_1, \dots, y_{k'}\} \to A$, defined by letting $\alpha(z_j) = \rho^{-1}(x_j)$ for all $1 \leq j \leq \mu_2$, and letting $\alpha(y_j) = c_{j'}$ where $1 \leq j \leq k'$ and where $1 \leq j' \leq u_2$ is the unique index so that applying ρ^{-1} to C'_j yields $C_{j'}$. It is then straightforward to verify that α witnesses that $\mathcal{A} \models \varphi_i$. Therefore, $\mathcal{A} \models \varphi$.

($\Leftarrow$) Conversely, suppose that $\mathcal{A} \models \varphi$. Then there must be some $1 \leq i \leq m_2$ such that $\mathcal{A} \models \varphi_i$. Suppose that φ_i is of the form $\exists z_1, \dots, z_{\mu_2} \exists y_1, \dots, y_{\mu_1} \psi$. Then there must be an assignment $\alpha : \{z_1, \dots, z_{\mu_2}\} \cup \{y_1, \dots, y_{\mu_1}\} \to A$ that witnesses that $\mathcal{A} \models \varphi_i$. Then, by construction of φ_i, we know that $\alpha(z_i) \in \{\, v_j : 1 \leq j \leq u_1 \,\}$ for all $1 \leq i \leq \mu_2$ and that $\alpha(y_i) \in \{\, c_j : 1 \leq j \leq u_2 \,\}$ for all $1 \leq i \leq \mu_1$. Then, consider the function $\iota : \{z_1, \dots, z_{\mu_2}\} \to \text{Var}(\mathcal{I})$, defined by letting $\iota(z_i) = x_j$ where j is the unique index such that $\alpha(z_i) = v_j$. Because $|\text{Var}(\mathcal{I}_i)| = \mu_2$, ι must be a bijection. Also, for each $1 \leq j \leq \mu_1$, let $(x_{\ell_{j,1}}, \dots, x_{\ell_{j,n_j}})$ be the scope of constraint $\alpha(y_j)$. Now, consider the set $\mathcal{I}' = \{C'_1, \dots, C'_{\mu_1}\}$ of constraints, where for each $1 \leq i \leq \mu_1$ the constraint C'_i has scope $S'_i = (\iota(z_{\ell_{j,1}}), \dots, \iota(z_{\ell_{j,n_j}})))$, and has as constraint relation $R'_i = R_\ell$, where $1 \leq \ell \leq m_3$ is the unique index such that the atom $T(y_i, \ell)$ occurs in φ_i. By construction of φ_i and $\mathcal{A}$, and by the choice of α, we get that $C'_i \in \mathcal{I}$ for each $1 \leq i \leq \mu_1$, and thus that $\mathcal{I}' \subseteq \mathcal{I}$. Moreover, the construction of $\mathcal{I}'$ gives us a mapping $\rho : \text{Var}(\mathcal{I}') \to \{x_1, \dots, x_{\mu_2}\}$ such that applying ρ to $\mathcal{I}'$ yields $\mathcal{I}_i$. (Namely, for each $x \in \text{Var}(\mathcal{I}')$, we let $\rho(x) = x_j$ where $1 \leq j \leq \mu_2$ is the unique index such that $\iota^{-1}(x) = z_j$.) Therefore, $\mathcal{I}'$ is unsatisfiable, and we can conclude that $(\mathcal{I}, k) \in$ SMALL-CSP-UNSAT-SUBSET. $\square$

4. BOOLEAN CONSTRAINT LANGUAGES

With the aim of identifying fragments for which finding small unsatisfiable subsets of a set of constraints, we turn our attention towards several fragments of Boolean constraints. By Proposition 3.1, Corollary 3.4 and Theorem 3.10, we know that merely restricting the domain to be Boolean does not lead to fixed-parameter tractability. There are, however, further restrictions on Boolean constraints that allow the satisfiability problem to be solved in polynomial time. Therefore, these are natural restrictions to investigate for our problem.

In his seminal paper, Schaefer [1978] identified the restrictions that need to be put on the individual constraints to result in a tractable satisfiability problem. These constraint-based restrictions are often characterized by the constraints (over a certain domain) that are allowed: the *constraint language*. In this section, we consider the problem of finding small unsatisfiable subsets for the six *Schaefer languages*: the Boolean constraint languages that Schaefer identified as the maximal languages that admit a polynomial-time satisfiability problem.

For these constraint languages, we study both the setting where constraints are specified as propositional formulas and the setting where constraints are specified as CSP instances. That is, we analyze the parameterized complexity of both problems SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET.

4.1. 0-Valid and 1-Valid Instances

We begin with 0-valid and 1-valid constraints. Since 0-valid and 1-valid constraints are always satisfiable (namely by the all-zeroes and the all-ones assignment, respectively), there are no unsatisfiable subsets when dealing with 0-valid or 1-valid constraints. This trivializes the problem of finding a small unsatisfiable subset.

OBSERVATION 4.1. SMALL-CSP-UNSAT-SUBSET *can be solved (trivially) in polynomial time when restricted to Boolean CSP instances that are* 0*-valid or* 1*-valid.* SMALL-CNF-UNSAT-SUBSET *can be solved (trivially) in polynomial time when restricted to CNF formulas that are* 0*-valid or* 1*-valid.*

4.2. Horn Instances

Next, we consider Horn and anti-Horn constraints. In this section, we work out results for Horn constraints, but all these results also hold for anti-Horn constraints: one can simply swap the domain values 0 and 1, or in terms of propositional formulas, negate all literals.

The reduction that Fellows et al. [2006, Section 4] use to show W[1]-hardness for SMALL-CNF-UNSAT-SUBSET in fact only uses clauses that are anti-Horn (that is, clauses that contain at most one negative literal). Therefore, this shows that the problem SMALL-CNF-UNSAT-SUBSET is W[1]-hard already when restricted to anti-Horn clauses (and thus is W[1]-hard also when restricted to Horn clauses).

PROPOSITION 4.2 ([FELLOWS ET AL. 2006]). SMALL-CNF-UNSAT-SUBSET *is* W[1]*-hard, even when restricted to Horn formulas.*

Fellows et al. also show hardness for SMALL-CNF-UNSAT-SUBSET when restricted to 3CNF formulas that are Horn, that is, Horn formulas where each clause has size at most 3. We revisit and slightly extend their result. We do this by relating the problem to the problem of finding small (directed) hyperpaths, with the aim of providing a bit more intuition behind the hardness result. In particular, we define the parameterized problem SHORT-HYPERPATH, show that it is W[1]-hard, and then provide an fpt-reduction from SHORT-HYPERPATH to the problem SMALL-CNF-UNSAT-SUBSET restricted to CNF formulas containing only Horn clauses of size at most 3.

For a Horn formula φ and $s, t \in \text{Var}(\varphi)$, we say that a subformula $\varphi' \subseteq \varphi$ is a *hyperpath* from s to t if (i) $t = s$ or (ii) $c = \{x_1, \dots, x_n, t\} \in \varphi'$ and $\varphi' \backslash c$ is a hyperpath from s to x_i for each $1 \leq i \leq n$. If $|\varphi'| \leq k$ then φ' is called a k-*hyperpath*. The parameterized problem SHORT-HYPERPATH takes as input a Horn formula φ, two variables $s, t \in \text{Var}(\varphi)$ and an integer k. The problem is parameterized by k. The question is whether there exists a k-hyperpath from s to t. For a more detailed discussion on the relation between (backward) hyperpaths in hypergraphs and hyperpaths as defined above, we refer to a survey article by Gallo et al. [1993]. For the hardness proof of SHORT-HYPERPATH, we reduce from the W[1]-complete problem MULTI-COLORED-CLIQUE [Fellows et al. 2009].

LEMMA 4.3. SHORT-HYPERPATH *is* W[1]-*hard, even for instances* (φ, s, t, k) *where* φ *contains only clauses of size at most* 3.

PROOF. We give a reduction from MULTI-COLORED-CLIQUE. Let (V, E, k) be an instance of MULTI-COLORED-CLIQUE, where $G = (V, E)$ is a simple graph and $V_1, \dots, V_k$ are the given partitions of V. We construct an instance (φ, s, t, k') of SHORT-HYPERPATH, where $k' = k + \binom{k}{2} + 1$ and

$$\begin{aligned}
\text{Var}(\varphi) &= \{s, t\} \cup V \cup \{\, p_{i,j} : 1 \leq i < j \leq k \,\}; \\
\varphi &= \varphi_V \cup \varphi_p \cup \varphi_t; \\
\varphi_V &= \{\, \{\neg s, v\} : v \in V \,\}; \\
\varphi_p &= \{\, \{\neg v_i, \neg v_j, p_{i,j}\} : 1 \leq i < j \leq k, v_i \in V_i, v_j \in V_j, \{v_i, v_j\} \in E \,\}; \\
\varphi_t &= \{\{\, \neg p_{i,j} : 1 \leq i < j \leq k \,\} \cup \{t\}\}.
\end{aligned}$$

This construction is illustrated for an example with $k = 3$ in Figure 1. We claim that $(G, k, c) \in$ MULTI-COLORED-CLIQUE if and only if $(\varphi, s, t, k') \in$ SHORT-HYPERPATH.

($\Rightarrow$) Assume $(G, k, c) \in$ MULTI-COLORED-CLIQUE. Then there exists a clique V' of G with $|V \cap V_i| = 1$ for all $1 \leq i \leq k$. We construct a k'-hyperpath φ' from s to t. We define:

$$\begin{aligned}
\varphi_{V'} = {} & \{\, \{\neg s, v\} : v \in V' \,\} \cup \varphi_t \cup \\
& \{\, \{\neg v_i, \neg v_j, p_{i,j}\} : 1 \leq i < j \leq k, v_i \in V_i \cap V', v_j \in V_j \cap V', \{v_i, v_j\} \in E \,\}
\end{aligned}$$

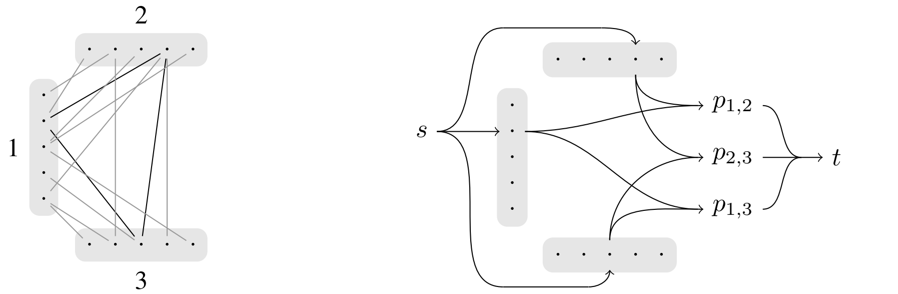
It is straightforward to verify that $\varphi_{V'}$ is a k'-hyperpath from s to t.

($\Leftarrow$) Assume that $(\varphi, s, t, k') \in$ SHORT-HYPERPATH. Then there exists a k'-hyperpath φ' from s to t. We know that $\varphi_t \subseteq \varphi'$, since φ_t contains the unique clause in φ with t occurring positively. Since $|\varphi'| \leq k'$, we know that in order for φ' to be a hyperpath from s to t, we have $|\varphi_V \cap \varphi'| = k$ and $|\varphi_p \cap \varphi'| = \binom{k}{2}$. It is then straightforward to verify that the set $V' = \{\, v \in V : \{\neg s, v\} \in \varphi' \,\}$ witnesses that G has a k-clique containing one node in each V_i.

To see that clauses of size at most 3 in the hyperpath suffice, we slightly adapt the reduction. The only clause we need to change is the single clause $e \in \varphi_t$. This clause e is of the form $\{\neg p_1, \dots, \neg p_m, t\}$, for $m = \binom{k}{2}$. We introduce new variables $z_1, \dots, z_m$ and replace e by the $m+1$ many clauses $\{\neg p_1, z_1\}$, $\{\neg z_{i-1}, \neg p_i, z_i\}$ for all $1 < i \leq m$ and $\{\neg z_m, t\}$. Clearly, the resulting Horn formula only has clauses of size at most 3. This adapted reduction works with the exact same line of reasoning as the reduction described above, with the only change that $k' = k + 2\binom{k}{2} + 1$.

Note that even the slightly stronger claim holds that G has a properly colored k-clique if and only if there exists a (subset) minimal k'-hyperpath $\varphi' \subseteq \varphi$ for which we have $|\varphi'| = k'$. □

We are now in a position to describe the proof that finding small unsatisfiable subsets in Horn formulas with clauses of size at most 3 is W[1]-hard. In fact, this proof shows that SMALL-CNF-UNSAT-SUBSET is W[1]-hard already when restricted to instances that contain only Horn clauses of size at most 3, where one clause is a unit clause containing a negative literal, and all other clauses are definite Horn (i.e., clauses that contain exactly one positive literal). Note that this hardness crucially depends on allowing one clause in the formula that is not definite Horn, since for definite Horn formulas the problem is trivial (definite Horn formulas are always satisfied by the truth assignment that sets all variables to 1). This hardness result for the case of Horn formulas containing clauses of



(a) A 3-partite graph G with a clique of size 3 (in black).

(b) The hyperpath in H of size $k' = 3 + \binom{3}{2} + 1$ from s to t corresponding to the clique.

Fig. 1: Illustration of the reduction in the proof of Lemma 4.3 for the case of a 3-colored clique.

size at most 3 was already shown by Fellows et al. [2006]. In fact, the fpt-reduction that they use to show this hardness result also only involves Horn formulas containing one clause c that is not definite Horn, but this clause c in their reduction is not a unit clause.

PROPOSITION 4.4. SMALL-CNF-UNSAT-SUBSET *is* W[1]*-hard even when restricted to the case where the input formula consists of Horn clauses of size at most* 3*, and contains only one clause that is not definite Horn.*

PROOF. We show W[1]-hardness by reducing from SHORT-HYPERPATH. Let (φ, s, t, k) be an instance of SHORT-HYPERPATH. By Lemma 4.3, we may assume without loss of generality that φ contains only definite Horn clauses of size at most 3. We construct an instance (ψ_φ, k') of SMALL-CNF-UNSAT-SUBSET. Here $k' = k + 2$. For each $\varphi' \subseteq \varphi$ we define a formula $\psi_{\varphi'}$, by letting $\text{Var}(\psi_{\varphi'}) = \text{Var}(\varphi')$ and:

$$\psi_{\varphi'} = \{\{s\}, \{\neg t\}\} \cup \varphi'.$$

Clearly ψ_φ contains only Horn clauses of size at most 3, and only the clause $\{\neg t\}$ is not definite Horn. We claim that $(\psi_\varphi, k') \in$ SMALL-CNF-UNSAT-SUBSET if and only if $(\varphi, s, t, k) \in$ SHORT-HYPERPATH.

($\Rightarrow$) Assume that there is some $\psi' \subseteq \psi_\varphi$ of size at most k that is unsatisfiable. Then $\{\neg t\} \in \psi'$, since otherwise setting all variables to 1 would satisfy ψ'. Similarly, also $\{s\} \in \psi'$, because otherwise setting all variables to 0 would satisfy ψ'. Now let $\varphi' \subseteq \varphi$ be the unique subset of clauses such that $\psi' = \psi_{\varphi'}$. We know that $|\varphi'| \leq k$. Now, since $\psi' \models s$ and $\psi' \models \neg t$, and because ψ' is unsatisfiable, we get that $\varphi' \models s \rightarrow t$. It is then easy to verify with an inductive argument that φ' is a k-hyperpath from s to t. In other words, $(\varphi, s, t, k) \in$ SHORT-HYPERPATH.

($\Leftarrow$) Assume that there exists a k-hyperpath $\varphi' \subseteq \varphi$ from s to t. By definition, $|\varphi'| \leq k$. It is readily verified with an inductive argument that this implies that $\varphi' \models s \rightarrow t$. Therefore, $\psi_{\varphi'} = \varphi' \cup \{\{s\}, \{\neg t\}\}$ is unsatisfiable. Since $|\psi_{\varphi'}| \leq k'$, this means that $(\psi_\varphi, k') \in$ SMALL-CNF-UNSAT-SUBSET. □

The above results directly give us W[1]-hardness for SMALL-CSP-UNSAT-SUBSET restricted to Horn constraints, because CNF formulas with bounded clause size can be expressed as equivalent CSP instances in polynomial time (this argument is similar to the one behind Corollary 3.14).

COROLLARY 4.5. SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances containing only Horn constraints (that are equivalent to a single Horn clause) is* W[1]*-hard.*

Alternatively, we can use the proof of Proposition 4.4 to show W[1]-hard for the problem SMALL-CNF-UNSAT-SUBSET restricted to Horn formulas containing only clauses of size at most 3 and containing only a single unit clause. Note that the problem is trivial for Horn formulas without unit clauses, since such formulas are always satisfiable (each clause then contains at least one negative literal, so setting all variables to 0 satisfies the formula).

COROLLARY 4.6. SMALL-CNF-UNSAT-SUBSET *is* W[1]*-hard even when restricted to instances* (φ, k) *where* φ *consists of Horn clauses of size at most* 3*, and contains only one unit clause.*

PROOF. We can modify the proof of Proposition 4.4 as follows. Instead of adding the unit clause $\{\neg t\}$, in the construction of the formulas $\psi_{\varphi'}$ we instantiate the formulas φ' with the literal $\neg t$ (and simplify the formula accordingly). Correspondingly, the new parameter value becomes $k' = k + 1$ (instead of $k + 2$). It is readily verified that this results in a correct fpt-reduction from SHORT-HYPERPATH to the problem SMALL-CNF-UNSAT-SUBSET, where each produced CNF formula satisfies the required constraints. □

For the case of CSP instances that are Horn, we can strengthen the W[1]-hardness results of Corollary 4.5 to a W[2]-hardness result by giving an fpt-reduction from HITTING-SET.

PROPOSITION 4.7. SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances containing only Horn constraints is* W[2]*-hard.*

PROOF. We show W[2]-hardness by reducing from HITTING-SET. Let $(U, \mathcal{T}, k)$ be an instance of HITTING-SET, where $\mathcal{T} = \{T_1, \dots, T_m\}$ is a family of subsets of the universe U. The idea behind this reduction is the following. We introduce a variable x_j for each subset $T_j \in \mathcal{T}$, plus an additional variable x_0. Then, we introduce a constraint C_0 that ensures that x_0 is assigned the value 1, and x_m is assigned the value 0. Moreover, for each element $u \in U$, we introduce a constraint C_u that entails the conjunction of the implications $(x_{j-1} \rightarrow x_j)$, for each T_j that u appears in. Any hitting set $H \subseteq U$ then entails to a conjunction of implications that together entail the implication $(x_0 \rightarrow x_m)$, which is inconsistent with the constraint C_0. Additionally, to ensure that for each constraint C_u the constraint relation R_u only contains a polynomial number of tuples, we add several Horn clauses to the CNF formula that is equivalent to the constraint C_u.

Formally, we construct the CSP instance $\mathcal{I}$ over the domain $D = \{0, 1\}$ as follows. We let Var$(\mathcal{I}) = \{x_0, \dots, x_m\}$. Then, we add the Horn constraint C_0 to $\mathcal{I}$, where:

$$C_0 \quad \equiv \quad x_0 \wedge \overline{x_m}.$$

Then, for each $u \in U$ we add a constraint C_u to $\mathcal{I}$. Take some $u \in U$, and let $T_{j_1}, \dots, T_{j_\ell}$ be the subsets $T_j \in \mathcal{T}$ that contain u, with $j_1 < \dots < j_\ell$. Then:

$$C_u \quad \equiv \quad \bigwedge_{1 \leq i \leq \ell} (x_{j_i - 1} \leftrightarrow x_{j_i}) \wedge \bigwedge_{1 \leq i < i' \leq \ell} (x_{j_i} \leftarrow x_{j_{i'}}).$$

Clearly, the constraint C_0 and all constraints C_u are Horn constraints. Also, the constraint relation R_0 (for the constraint C_0) and all constraint relations R_u (for the constraints C_u) contain a polynomial number of tuples. For the constraint C_0 this is clear. Consider some constraint $C_u = (S_u, R_u)$. Assuming that the order of the variables in the scope $S_u = (x_{i_1}, \dots, x_{i_\ell})$ is increasing (i.e., $i_1 < \dots < i_\ell$), the constraint relation R_u contains only tuples consisting of a number of ones (possibly zero of them) followed by a number of zeroes (possibly zero of them). These are only linearly many.

Finally, we let $k' = k + 1$. We claim that $(U, \mathcal{T}, k) \in$ HITTING-SET if and only if $(\mathcal{I}, k) \in$ SMALL-CSP-UNSAT-SUBSET.

$(\Rightarrow)$ Assume that there is a hitting set $H \subseteq U$ of $\mathcal{T}$ of size at most k. Take the set $\mathcal{I}' = \{C_0\} \cup \{ C_u : u \in H \} \subseteq \mathcal{I}$ of constraints. Clearly, $|\mathcal{I}'| \leq k'$. We show that $\mathcal{I}'$ is unsatisfiable, by showing that $\mathcal{I}' \models (x_{j-1} \rightarrow x_j)$ for each $1 \leq j \leq m$. Then, by transitivity, $\mathcal{I}' \models (x_0 \rightarrow x_m)$, and since $C_0 \in \mathcal{I}'$, we get that $\mathcal{I}'$ is unsatisfiable.

Take an arbitrary $1 \leq j \leq m$. Because H is a hitting set of $\mathcal{T}$, we know that there is some $u \in H \cap T_j$. By definition of C_u we know that $C_u \models (x_{j-1} \rightarrow x_j)$. Then, since $C_u \in \mathcal{I}'$, we know that $\mathcal{I}' \models (x_{j-1} \rightarrow x_j)$.

Therefore, we can conclude that $\mathcal{I}$ contains an unsatisfiable subset of size at most k'.

($\Leftarrow$) Conversely, assume that $\mathcal{I}$ contains an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size at most k'. We know that $C_0 \in \mathcal{I}'$, because the assignment that sets all variables to 1 satisfies each constraint C_u. Let $\mathcal{I}'' = \mathcal{I}' \setminus \{C_0\}$. We know that $|\mathcal{I}''| \leq k$. Now, consider the set $H = \{ u \in U : C_u \in \mathcal{I}'' \} \subseteq U$. We show that H is a hitting set of $\mathcal{T}$.

Take an arbitrary subset $T_j \in \mathcal{T}$. To derive a contradiction, suppose that $H \cap T_j = \emptyset$. We show that $\mathcal{I}'$ is then satisfiable. Consider the assignment α that sets the variables $x_0, x_1, \dots, x_{j-1}$ to 1, and all variables $x_j, \dots, x_m$ to 0. Clearly, α satisfies C_0. Moreover, since $H \cap T_j = \emptyset$ we know that $\mathcal{I}'$ contains no constraint C_u for which it holds that $C_u \models (x_{j-1} \rightarrow x_j)$. Therefore, α also satisfies all constraints $C_u \in \mathcal{I}'$. Thus, $\mathcal{I}'$ is satisfiable, which is a contradiction, and we can conclude that $H \cap T_j \neq \emptyset$. Since T_j was arbitrary, we conclude that H is a hitting set of $\mathcal{T}$ (of size at most k). $\square$

4.3. Bijunctive Instances

We now turn our attention to bijunctive constraints. For the case of bijunctive CNF formulas (that is, 2CNF or Krom formulas), Buresh-Oppenheim and Mitchell showed that a dynamic programming algorithm can be used to find a smallest unsatisfiable subset in polynomial time [Buresh-Oppenheim and Mitchell 2006, Section 4]. Therefore, this algorithm can be used to solve the problem SMALL-CNF-UNSAT-SUBSET in polynomial time for bijunctive CNF formulas.

PROPOSITION 4.8 ([BURESH-OPPENHEIM AND MITCHELL 2006]). SMALL-CNF-UNSAT-SUBSET *restricted to bijunctive formulas can be solved in polynomial time.*

Their algorithm critically uses the fact that a bijunctive clause $\{l_1, l_2\}$ corresponds to exactly two implications: $\neg l_1 \rightarrow l_2$ and $\neg l_2 \rightarrow l_1$. For the case of bijunctive Boolean CSP instances, this is not the case. To see this, consider a bijunctive constraint that is equivalent to the bijunctive CNF formula consisting of the two clauses $\{a, b\}, \{c, d\}$. Then this constraint corresponds to four implications: $\neg a \rightarrow b$, $\neg b \rightarrow a$, $\neg c \rightarrow d$, and $\neg d \rightarrow c$. As a result, the polynomial-time algorithm by Buresh-Oppenheim and Mitchell does not work for the case of CSP instances.

In fact, the problem SMALL-CSP-UNSAT-SUBSET restricted to bijunctive Boolean CSP instances is W[1]-hard, even when the constraints have arity at most 4. By using the fact that a single bijunctive constraint can encode multiple implications, we are able to give an fpt-reduction from MULTI-COLORED-CLIQUE.

PROPOSITION 4.9. SMALL-CSP-UNSAT-SUBSET *restricted to bijunctive Boolean CSP instances, where each constraint is equivalent to at most* 4 *bijunctive clauses, is* W[1]*-hard.*

PROOF. We provide an fpt-reduction from MULTI-COLORED-CLIQUE. Let (G, k) be an instance of MULTI-COLORED-CLIQUE, where $G = (V, E)$ and where V is partitioned into $V_1, \dots, V_k$. We construct a bijunctive Boolean CSP instance $\mathcal{I}$, and an integer k'. We let Var($\mathcal{I}$) consist of variables $x_{v,j}$ for each $v \in V$, each $1 \leq j \leq k+1$, plus a variable z_0. We then let $\mathcal{I}$ consist of the following constraints. For each $e = (v, v') \in (V_i \times V_j) \cap E$, for $i < j$, we introduce the constraint $C_e = (S_e, R_e)$, where $S_e = (x_{v,j}, x_{v,j+1}, x_{v',i}, x_{v',i+1})$ and where R_e is equivalent to the following Krom formula:

$$\begin{aligned} R_e \equiv\ & (x_{v,j} \leftrightarrow x_{v,j+1}) \wedge (x_{v',i} \leftrightarrow x_{v',i+1}) \\ \equiv\ & (x_{v,j} \rightarrow x_{v,j+1}) \wedge (x_{v,j+1} \rightarrow x_{v,j}) \wedge \\ & (x_{v',i} \rightarrow x_{v',i+1}) \wedge (x_{v',i+1} \rightarrow x_{v',i}). \end{aligned}$$

For each $1 \leq i \leq k$, and each $v \in V_i$, we introduce the constraint $C_v = (S_v, R_v)$, where $S_v = (x_{v,i}, x_{v,i+1})$ and where R_v is equivalent to the following Krom formula:

$$\begin{aligned} R_v \equiv\ & (x_{v,i} \leftrightarrow x_{v,i+1}) \\ \equiv\ & (x_{v,i} \rightarrow x_{v,i+1}) \wedge (x_{v,i+1} \rightarrow x_{v,i}). \end{aligned}$$

Then, for each $1 \leq i < k$, each $v \in V_i$ and each $v' \in V_{i+1}$, we introduce the constraint $C_{v,v'} = (S_{v,v'}, R_{v,v'})$, where $S_{v,v'} = (x_{v,k+1}, x_{v',1})$ and where $R_{v,v'}$ is equivalent to the following Krom formula:

$$\begin{aligned} R_{v,v'} \equiv\ & (x_{v,k+1} \leftrightarrow x_{v',1}) \\ \equiv\ & (x_{v,k+1} \rightarrow x_{v',1}) \wedge (x_{v',1} \rightarrow x_{v,k+1}). \end{aligned}$$

Finally, for each $v \in V_1$ and each $v' \in V_k$, we introduce the constraint $C_{v,v'} = (S_{v,v'}, R_{v,v'})$, where $S_{v,v'} = (z_0, x_{v,1}, x_{v',k+1})$ and where $R_{v,v'}$ is equivalent to the following Krom formula:

$$\begin{aligned} R_{v,v'} \equiv\ & (z_0 \leftrightarrow x_{v,1}) \wedge (x_{v',k+1} \leftrightarrow \overline{z_0}) \\ \equiv\ & (z_0 \rightarrow x_{v,1}) \wedge (x_{v,1} \rightarrow z_0) \wedge \\ & (x_{v',k+1} \rightarrow \overline{z_0}) \wedge (\overline{z_0} \rightarrow x_{v',k+1}). \end{aligned}$$

Note that the scope of each constraint is of constant size, so the constraints are all of constant size when spelled out. Finally, we let $k' = \binom{k}{2} + 2k$.

We claim that $(G, k) \in$ MULTI-COLORED-CLIQUE if and only if there exists some subset $\mathcal{I}' \subseteq \mathcal{I}$ of k' many constraints that is unsatisfiable. The intuition behind this construction is the following. Any unsatisfiable subset needs to force z_0 to be true and false at the same time. This can only be done with a chain of equivalences. Any chain of equivalences with this property that is represented by at most k' many constraints corresponds to a multi-colored k-clique in G.

($\Rightarrow$) Assume that G has a multi-colored k-clique, i.e., there exists some set $\{v_{\ell_1}, \dots, v_{\ell_k}\} \subseteq V$ of vertices such that for each $1 \leq i \leq k$, $v_{\ell_i} \in V_i$, and for each $1 \leq i < j \leq k$, $(v_{\ell_i}, v_{\ell_j}) \in E$. Consider the subset $\mathcal{I}' \subseteq \mathcal{I}$ consisting of the following constraints:

$$\begin{aligned} \mathcal{I}' =\ & \{\, C_e : 1 \leq i < j \leq k, e = (v_{\ell_i}, v_{\ell_j}) \,\} \cup \\ & \{\, C_{v_{\ell_i}} : 1 \leq i \leq k \,\} \cup \\ & \{\, C_{v_{\ell_i}, v_{\ell_j}} : 1 \leq i \leq k, j = i+1 \pmod{k} \,\}. \end{aligned}$$

It is easy to verify that $\mathcal{I}'$ consists of k' many constraints. Moreover, it is straightforward to verify that any solution α of $\mathcal{I}'$ must satisfy that $\alpha(z_0) = \alpha(\overline{z_0})$. Thus, $\mathcal{I} \in$ SMALL-CSP-UNSAT-SUBSET.

($\Leftarrow$) Conversely, assume that there is some inconsistent subset $\mathcal{I}' \subseteq \mathcal{I}$ of at most k' many constraints. We show that $(G, k) \in$ MULTI-COLORED-CLIQUE. We know that $\mathcal{I}'$ must include the constraint $C_{v,v'}$, for some $v_{\ell_1} \in V_1$ and some $v_{\ell_k} \in V_k$. Otherwise, the assignment setting all variables to 1 would satisfy $\mathcal{I}'$. Moreover, we know that $\mathcal{I}'$ must include a sequence of constraints that together enforce the equivalence $(x_{v_{\ell_1},1} \leftrightarrow x_{v_{\ell_k},k+1})$; otherwise $\mathcal{I}'$ would be satisfiable. Then we also know that there exist vertices $v_{\ell_2}, \dots, v_{\ell_{k-1}}$ such that for each $2 \leq j < k$, $v_{\ell_j} \in V_j$ and such that $\mathcal{I}'$ includes, for each $1 \leq i < k$, the constraint $C_{v_{\ell_i}, v_{\ell_{i+1}}}$; otherwise, the equivalence $(x_{v_{\ell_1},1} \leftrightarrow x_{v_{\ell_k},k+1})$ would not be enforced. Finally, $\mathcal{I}'$ must enforce the equivalences $(x_{v_{\ell_i},1} \leftrightarrow x_{v_{\ell_i},k+1})$, for each $1 \leq i \leq k$. It is straightforward to verify that the only way to do this with $k + \binom{k}{2}$ additional constraints, is to choose $C_{v_{\ell_i}}$ for each $1 \leq i \leq k$, and the constraints $C_{e_{ij}}$ for each $e_{ij} = (v_{\ell_i}, v_{\ell_j})$, for $1 \leq i < j \leq k$. If such constraints $C_{e_{ij}}$ are present, then clearly, by construction of the set $\mathcal{I}$, the set $\{v_{\ell_1}, \dots, v_{\ell_k}\}$ is a multi-colored k-clique of G. Therefore, $(G, k) \in$ MULTI-COLORED-CLIQUE. $\square$

In the proof of the above W[1]-hardness result, only constraint of arity at most 4 are used. In the more general case where (bijunctive) constraints of unbounded arity are allowed, the problem is even W[2]-hard.

COROLLARY 4.10. SMALL-CSP-UNSAT-SUBSET *restricted to bijunctive Boolean CSP instances is* W[2]*-hard.*

PROOF. In the reduction in the proof of Proposition 4.7, all contraints are bijunctive. Therefore, this is also an fpt-reduction from HITTING-SET to SMALL-CSP-UNSAT-SUBSET restricted to bijunctive instances. □

4.4. Affine Instances

In the proof of Proposition 4.9, when showing W[1]-hardness for SMALL-CSP-UNSAT-SUBSET restricted to bijunctive constraints, we only used bijunctive constraints that are equivalent to conjunctions of equivalences. Since such conjunctions of equivalences can also be expressed as 2-affine constraints, we can straightforwardly transfer the W[1]-hardness result to the case of affine constraints.

COROLLARY 4.11. SMALL-CSP-UNSAT-SUBSET *restricted to 2-affine Boolean CSP instances is* W[1]*-hard.*

PROOF. The result follows from the fpt-reduction in the proof of Proposition 4.9. All propositional formulas used to define the constraints of the resulting CSP instance are conjunctions of equivalences of the form $(l_1 \leftrightarrow l_2)$. Each such equivalence can be expressed by the affine clause $\neg(l_1 \oplus l_2) \equiv (\overline{l_1} \oplus l_2)$, containing only two literals. Therefore the resulting CSP instance is also a 2-affine Boolean CSP instance. □

The above result does not suffice to indicate whether or not SMALL-CNF-UNSAT-SUBSET is fixed-parameter tractable when the input formulas are affine, because the proofs of Proposition 4.9 and Corollary 4.11 depend on constraints that are equivalent to a conjunction of more than one (affine) clauses. The next result shows that SMALL-CSP-UNSAT-SUBSET is even W[1]-hard when restricted to affine constraints that are equivalent to a single affine clause. This will allow us to transfer the hardness result to the case of affine formulas.

PROPOSITION 4.12. SMALL-CSP-UNSAT-SUBSET *is* W[1]*-hard when restricted to Boolean CSP instances where each constraint is equivalent to a single affine clause.*

PROOF. We provide an fpt-reduction from MULTI-COLORED-CLIQUE. This reduction is similar to the reduction given in the proof of Proposition 4.9. Let (G, k) be an instance of MULTI-COLORED-CLIQUE, where $G = (V, E)$ and where V is partitioned into $V_1, \dots, V_k$. We construct a bijunctive Boolean CSP instance $\mathcal{I}$, and an integer k'. We let Var$(\mathcal{I})$ consist of variables $x_{v,j}$ for each $v \in V$, each $1 \leq j \leq k+1$, and variables z_0, z_1. We then let $\mathcal{I}$ consist of the following constraints. For each $e = (v, v') \in (V_i \times V_j) \cap E$, for $1 \leq i < j \leq k$, we introduce the constraint $C_e = (S_e, R_e)$, where $S_e = (x_{v,j}, x_{v,j+1}, x_{v',i}, x_{v',i+1})$ and where R_e is equivalent to the following affine clause:

$$\begin{aligned} R_e \equiv\ & \neg(x_{v,j} \oplus x_{v,j+1} \oplus x_{v',i} \oplus x_{v',i+1}) \\ \equiv\ & (\overline{x_{v,j}} \oplus x_{v,j+1} \oplus x_{v',i} \oplus x_{v',i+1}). \end{aligned}$$

For each $1 \leq i \leq k$, and each $v \in V_i$, we introduce the constraint $C_v = (S_v, R_v)$, where $S_v = (x_{v,i}, x_{v,i+1})$ and where R_v is equivalent to the following affine clause:

$$R_v \equiv \neg(x_{v,i} \oplus x_{v,i+1}) \equiv (\overline{x_{v,i}} \oplus x_{v,i+1}).$$

Then, for each $1 \leq i < k$, each $v \in V_i$ and each $v' \in V_{i+1}$, we introduce the constraint $C_{v,v'} = (S_{v,v'}, R_{v,v'})$, where $S_{v,v'} = (x_{v,k+1}, x_{v',1})$ and where $R_{v,v'}$ is equivalent to the following affine clause:

$$R_{v,v'} \equiv \neg(x_{v,k+1} \oplus x_{v',1}) \equiv (\overline{x_{v,k+1}} \oplus x_{v',1}).$$

Finally, for each $v \in V_1$ and each $v' \in V_k$, we introduce the constraint $C_{v,v'} = (S_{v,v'}, R_{v,v'})$, where $S_{v,v'} = (z_0, x_{v,1}, x_{v',k+1})$ and where $R_{v,v'}$ is equivalent to the following affine clause:

$$\begin{aligned} R_{v,v'} &\equiv \neg(z_0 \oplus x_{v,1} \oplus x_{v',k+1} \oplus z_1) \\ &\equiv (\overline{z_0} \oplus x_{v,1} \oplus x_{v',k+1} \oplus z_1). \end{aligned}$$

Finally, we add the constraint $C_0 = (S_0, R_0)$, where $S_0 = (z_0, z_1)$ and where R_0 is equivalent to the affine clause $R_0 \equiv (z_0 \oplus z_1)$. Note that the scope of each constraint is of constant size, so the constraints are all of constant size when spelled out. Finally, we let $k' = \binom{k}{2} + 2k + 1$.

The intuition behind this construction is the following. Any unsatisfiable subset needs to force z_0 to be assigned the same value as z_1, which then leads to unsatisfiability due to the constraint C_0. This can only be done with a path from z_0 to z_1 in the incidence graph of the instance. The *incidence graph* of an instance $\mathcal{I}$ is the graph $G_{\mathcal{I}} = (\text{Var}(\mathcal{I}'), E_{\mathcal{I}'})$, where $\{x, x'\} \in E_{\mathcal{I}'}$ if and only if x and x' occur together in some clause $C \in \mathcal{I}'$. Any such path corresponding to at most k' many constraints corresponds to a multi-colored k-clique in G.

We claim that $(G, k) \in$ MULTI-COLORED-CLIQUE if and only if there exists some subset $\mathcal{I}' \subseteq \mathcal{I}$ of k' many constraints that is unsatisfiable.

($\Rightarrow$) Assume that G has a multi-colored k-clique, i.e., there exists some set $\{v_{\ell_1}, \dots, v_{\ell_k}\} \subseteq V$ of vertices such that for each $1 \leq i \leq k$, $v_{\ell_i} \in V_i$, and for each $1 \leq i < j \leq k$, $(v_{\ell_i}, v_{\ell_j}) \in E$. Consider the subset $\mathcal{I}' \subseteq \mathcal{I}$ consisting of the following constraints:

$$\begin{aligned} \mathcal{I}' = \; & \{\, C_e : 1 \leq i < j \leq k, e = (v_{\ell_i}, v_{\ell_j}) \,\} \cup \\ & \{\, C_{v_{\ell_i}} : 1 \leq i \leq k \,\} \cup \{C_0\} \cup \\ & \{\, C_{v_{\ell_i}, v_{\ell_j}} : 1 \leq i \leq k, j = i+1 \pmod{k} \,\}. \end{aligned}$$

It is easy to verify that $\mathcal{I}'$ consists of k' many constraints. Moreover, consider the following sequence of literals over variables in Var($\mathcal{I}'$).

$$\begin{aligned} \sigma = \; & (l_1, \dots, l_m) \\ = \; & (z_0, x_{v_{\ell_1},1}, \dots, x_{v_{\ell_1},k+1}, x_{v_{\ell_2},1}, \dots, x_{v_{\ell_k},k+1}, z_1). \end{aligned}$$

The reader can easily verify that (i) each pair (l_i, l_{i+1}) of literals, for $1 \leq i < m$, occurs in exactly one constraint in $\mathcal{I}' \backslash \{C_0\}$, (ii) that each literal l_i, for $1 < i < m$, occurs in exactly two constraints in $\mathcal{I}' \backslash \{C_0\}$, and (iii) that the literals l_1 and l_m each occur in exactly one constraint in $\mathcal{I}' \backslash \{C_0\}$. Due to these properties, the constraints in $\mathcal{I}' \backslash \{C_0\}$ together enforce that for any solution α of $\mathcal{I}'$ there must be an even number of indices $1 \leq i < m$ such that $\alpha(l_i) \neq \alpha(l_{i+1})$. This entails that for any solution α of $\mathcal{I}'$ it must hold that $\alpha(z_0) = \alpha(z_1)$. However, since $C_0 \in \mathcal{I}'$, we get that $\mathcal{I} \in$ SMALL-CSP-UNSAT-SUBSET.

($\Leftarrow$) Conversely, assume that there is some inconsistent subset $\mathcal{I}' \subseteq \mathcal{I}$ of at most k' many constraints. We show that $(G, k) \in$ MULTI-COLORED-CLIQUE. We know that $\mathcal{I}'$ must include the constraint C_0. Otherwise, the assignment setting all variables to 1 would satisfy $\mathcal{I}'$.

Next, we consider the incidence graph $G_{\mathcal{I}'}$ of $\mathcal{I}'$. We then know that there must be a path in $G_{\mathcal{I}'}$ from z_0 to z_1. Otherwise $\mathcal{I}'$ would be satisfiable; a solution for $\mathcal{I}'$ would be the assignment that sets all variables connected in $G_{\mathcal{I}'}$ to z_0 to the value 0, and all other variables to the value 1.

By an argument similar to the one in the proof of Proposition 4.9, we then know that if such a path can be constructed with $k' - 1$ many constraints (in addition to C_0), then G must contain a multi-colored k-clique, and thus $(G, k) \in$ MULTI-COLORED-CLIQUE. $\square$

As a corollary, we get W[1]-hardness for the problem SMALL-CNF-UNSAT-SUBSET restricted to affine formulas, that is, conjunctions of affine clauses.

COROLLARY 4.13. SMALL-CNF-UNSAT-SUBSET *is* W[1]-*hard when input formulas are affine formulas.*

PROOF. The instances constructed in the fpt-reduction in the proof of Proposition 4.12 only contain affine constraints that are equivalent to a single affine clause. Therefore, we can readily rephrase it as an fpt-reduction from MULTI-COLORED-CLIQUE to SMALL-CNF-UNSAT-SUBSET with affine formulas as input. □

5. INSTANCES WITH BOUNDED VARIABLE OCCURRENCE

In this section, we consider another restriction for the problems SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET. Namely, we bound the maximum number of times that any variable occurs in instances. In the case of SMALL-CNF-UNSAT-SUBSET, this restriction directly leads to fixed-parameter tractability. For the case of SMALL-CSP-UNSAT-SUBSET, it turns out that we only get fixed-parameter tractability when further restrictions are made in addition (bounding the arity of constraints and the domain size).

5.1. CNF Formulas

The fixed-parameter tractability result for the case of CNF formulas with bounded degree is in fact already implied by the result that SMALL-CNF-UNSAT-SUBSET is fixed-parameter tractable for instances restricted to classes of formulas that have locally bounded treewidth [Fellows et al. 2006]. Fellows et al. used a meta theorem to prove this. We give a direct (bounded search tree) algorithm to solve SMALL-CNF-UNSAT-SUBSET in fixed-parameter linear time for instances with bounded degree.

We describe the algorithm as a non-deterministic algorithm, and then argue that simulating all non-deterministic choices can be done in fixed-parameter linear time. Let (φ, k) be an instance of SMALL-CNF-UNSAT-SUBSET with degree d. The following procedure decides whether there exists an unsatisfiable subset $\varphi' \subseteq \varphi$ of size at most k, and computes such a subset if it exists. We let $\varphi^\star = \{ c \in \varphi : |c| < k \}$. It suffices to consider subsets of $\varphi^\star$, since any unsatisfiable subset $\varphi' \subseteq \varphi$ contains a minimally unsatisfiable subset $\varphi'' \subseteq \varphi'$, and by Tarsi's Lemma we know that φ'' contains only clauses of size smaller than k.

Without loss of generality, we assume that the incidence graph of $\varphi^\star$ is connected. Otherwise, we can solve the problem by running the algorithm on each of (the subsets induced by) the connected components. We guess a clause $c \in \varphi^\star$, we let $F_1 := \{c\}$, and we let all variables be unmarked initially. We compute F_{i+1} for $1 \leq i < k$ by means of the following (non-deterministic) rule:

(1) take an unmarked variable $z \in \text{Var}(F_i)$;
(2) guess a non-empty subset $G'_z \subseteq G_z$ for $G_z = \{ c \in \varphi^\star : z \in \text{Var}(c) \}$;
(3) let $F_{i+1} := F_i \cup G'_z$;
(4) mark z.

If at any point all variables in F_i are marked, we stop computing F_{i+1}. For any F_i, if $|F_i| > k$, we fail. For each F_i, we check whether F_i is unsatisfiable. If it is unsatisfiable, we return with $\varphi' = F_i$. If it is satisfiable and if it contains no unmarked variables, we fail. Otherwise (that is, if it is satisfiable and contains unmarked variables), we continue with computing F_{i+1}.

It is easy to see that this algorithm is sound. If some $\varphi' \subseteq \varphi^\star$ is returned, then φ' is unsatisfiable and $|\varphi'| \leq k$. In order to see that the algorithm is complete, assume that there exists some unsatisfiable $\varphi' \subseteq \varphi^\star$ with $|\varphi'| \leq k$. Then, since we know that the incidence graph of φ' is connected, we know that φ' can be constructed as one of the F_i in the algorithm.

To see that this algorithm witnesses fixed-parameter linearity, we bound its running time. We have to execute the search process at most once for each clause of $\varphi^\star$. At each point in the execution of the algorithm, F_i contains at most k variables. Therefore, there are at most k choices to take an unmarked variable z. Since each variable occurs in at most d clauses, for each G_z used in the rule we know $|G_z| \leq d$. Thus, there are at most 2^d possible guesses for G'_z in each execution of the rule. Since we iterate the rule at most k times, we consider at most $(k2^d)^k$ sets F_i, each of size $O(k^2)$. Since each set F_i contains at most k variables, each (un)satisfiability check can be done in time $O(2^k)$. Therefore, the total running time of the algorithm is $O(k^k 2^{dk} n)$, where n is the size of the instance.

PROPOSITION 5.1. SMALL-CNF-UNSAT-SUBSET *is fixed-parameter linear when restricted to CNF formulas of degree at most* $f_1(k)$, *where* f_1 *is a computable function and* k *is the parameter value.*

PROOF. The result follows directly by using the above algorithm, where we let $d = f_1(k)$. □

5.2. CSP Instances

We consider a similar restriction for CSP instances (instances with bounded degree). Remember that the degree of a CSP instance $\mathcal{I}$ is defined as the maximum number of constraints $C \in \mathcal{I}$ that any variable $x \in \text{Var}(\mathcal{I})$ appears in. For the problem SMALL-CSP-UNSAT-SUBSET, bounding only the degree does not lead to fixed-parameter tractability.

Only bounding the degree of CSP instances yields fixed-parameter intractability. In particular, for any constant $d \geq 2$, the problem SMALL-CSP-UNSAT-SUBSET is already W[1]-hard when restricted to Boolean CSP instances whose degree is bounded by d. (For CSP instances with degree 1, the problem SMALL-CSP-UNSAT-SUBSET is trivial, since different constraints cannot share variables.) This result implies that the problem is also fixed-parameter intractable when both the degree and the domain size of the CSP instances are bounded.

PROPOSITION 5.2. SMALL-CSP-UNSAT-SUBSET *is* W[1]-*hard, even when restricted to Boolean CSP instances with degree* 2.

PROOF. We know that the problem of finding an unsatisfiable subset of a 3CNF formula of size at most k is W[1]-hard. We give an fpt-reduction from this problem to SMALL-CSP-UNSAT-SUBSET. The idea behind this reduction is to introduce many copies of each variable (one copy for each occurrence) and to introduce for each variable a single constraint that ensures that all copies of this variable are assigned the same value.

Let $\varphi = \{c_1, \dots, c_m\}$ be a propositional formula in 3CNF, and let k be a positive integer. By Lemma 4.3 and Proposition 4.4, we may assume without loss of generality that φ contains no unsatisfiable subsets of size strictly less than k, and that any unsatisfiable subset of φ of size k contains exactly $\ell = k - 1$ variables.

We now construct an instance $(\mathcal{I}, k')$ of SMALL-CSP-UNSAT-SUBSET as follows. We let $\text{Var}(\mathcal{I}) = \{\, v_{x,c} : x \in \text{Var}(\varphi), c \in \varphi \,\}$, and we let $D = \{0, 1\}$. Then, for each $c \in \varphi$, we add a constraint $C_c = (S_c, R_c)$ to $\mathcal{I}$. Let $c = (l_x \vee l_y \vee l_z)$, where l_x is a literal over variable x, l_y a literal over y and l_z a literal over z. We let $S_c = (v_{x,c}, v_{y,c}, v_{z,c})$, and we define R_c to be set of 3-tuples $\overline{s} \in \{0, 1\}^3$ satisfying c. Next, for each variable $x \in \text{Var}(\varphi)$, we add a constraint $C_x = (S_x, R_x)$ to $\mathcal{I}$. We let $S_x = (v_{x,c_1}, \dots, v_{x,c_m})$, and we let $R_x = \{(0, \dots, 0), (1, \dots, 1)\}$. Finally, we define $k' = k + \ell$. It is straightforward to verify that $\mathcal{I}$ has degree 2. We now show that φ has an unsatisfiable subset of size k if and only if $\mathcal{I}$ has an unsatisfiable subset of size k'.

($\Rightarrow$) Assume that φ has an unsatisfiable subset of size k. Let $\varphi' = \{c'_1, \dots, c'_k\}$ be such a subset. We know that exactly ℓ variables appear in φ'. Now consider the set $\mathcal{I}' = \{\, C_{c'_i} : 1 \leq i \leq k \,\} \cup \{\, C_x : x \in \text{Var}(\varphi') \,\}$ of constraints. It is straightforward to verify that $\mathcal{I}'$ is an unsatisfiable subset of $\mathcal{I}$ containing k' many constraints.

($\Leftarrow$) Conversely, assume that $\mathcal{I}$ has an unsatisfiable subset of size at most k'. Let $\mathcal{I}'$ be such a subset, and let $\mathcal{I}'$ be minimal. Then, let $\mathcal{I}'_1 = \mathcal{I}' \cap \{\, C_c : c \in \varphi \,\}$ and let $\mathcal{I}'_2 = \mathcal{I}' \cap \{\, C_x : x \in \text{Var}(\varphi) \,\}$. Then let $k_1 = |\mathcal{I}'_1|$ and let $k_2 = |\mathcal{I}'_2|$. We show that $k_1 = k$ and $k_2 = \ell$. We proceed indirectly. Firstly, suppose that $k_1 < k$. It is then straightforward to verify that $\varphi' = \{\, c \in \varphi : C_c \in \mathcal{I}'_1 \,\}$ is an unsatisfiable subset of φ of size $< k$, which is a contradiction. Thus, $k_1 \geq k$. Next, suppose that $k_2 < \ell$. We know that φ' is an unsatisfiable subset of φ. However, by minimality of $\mathcal{I}'$, we then know that φ' contains $k_2 < \ell$ many variables, which is a contradiction. Now, since $k' = k + \ell \geq k_1 + k_2$, we can conclude that $k_1 = k$ and $k_2 = \ell$. Therefore, φ' is an unsatisfiable subset of φ of size k. □

Similarly, bounding the degree of the CSP instance and the arity of constraints (but allowing unbounded domain size) leads to hardness. We show that the problem SMALL-CSP-UNSAT is co-W[1]-hard, even when restricted to CSP instances with degree 3 and maximum arity 2. This directly gives us co-W[1]-hardness also for SMALL-CSP-UNSAT-SUBSET.

PROPOSITION 5.3. SMALL-CSP-UNSAT *is* co-W[1]-*hard, even when restricted to CSP instances with degree* 3 *and maximum arity* 2.

PROOF. We provide an fpt-reduction from co-CLIQUE Let (G, k) be an instance of co-CLIQUE, where $G = (V, E)$ is a graph. We construct a CSP instance $\mathcal{I}$ containing $k' = k(k-1) + \binom{k}{2}$ constraints, such that $\mathcal{I}$ is satisfiable if and only if G has a k-clique.

We let $\text{Var}(\mathcal{I}) = \{ x_j^i : 1 \leq i \leq k, 1 \leq j \leq k \}$, and we let the domain of $\mathcal{I}$ be $D = V$. Then, for each $1 \leq i \leq k$ and each $1 \leq j < k$ we introduce a constraint $C_{i,j}^{\text{succ}}$, with $\text{Var}(C_{i,j}^{\text{succ}}) = \{x_i^j, x_i^{j+1}\}$. The constraint relation encodes equality on V, that is, $R_{i,j}^{\text{succ}} = \{ (v, v) : v \in V \}$.

Also, for each $1 \leq i < j \leq k$, we introduce a constraint $C_{i,j}^E$, with $\text{Var}(C_{i,j}^E) = \{x_i^j, x_j^i\}$. The constraint relation encodes the edge set E. We let $R_{i,j}^E = \{ (v, w) : v, w \in V, \{v, w\} \in E \}$.

Clearly, $\mathcal{I}$ contains k' many constraints. Also, the maximum arity of $\mathcal{I}$ is 2. To see that the degree of $\mathcal{I}$ is 3, take an arbitrary variable x_i^j. The only constraints that x_i^j can appear in are either (1) of the form $C_{i,j}^{\text{succ}}$ (if $j < k$), or (2) of the form $C_{i,j-1}^{\text{succ}}$ (if $j > 1$), or (3) of the form $C_{i,j}^E$ (if $i < j$) or (4) of the form $C_{j,i}^E$ (if $i > j$).

To show that this reduction is correct, we show that $(G, k) \in$ CLIQUE if and only if $\mathcal{I}$ is satisfiable.

($\Rightarrow$) Take a k-clique $V' = \{v_1, \dots, v_k\} \subseteq V$. We construct the following assignment $\alpha : \text{Var}(\mathcal{I}) \to D$. We let $\alpha(x_i^j) = v_i$, for each $1 \leq i \leq k$ and each $1 \leq j \leq k$. Using the fact that V' is a clique, it is readily verified that α satisfies all constraints in $\mathcal{I}$.

($\Leftarrow$) Conversely, take an assignment $\alpha : \text{Var}(\mathcal{I}) \to D$ that satisfies all constraints in $\mathcal{I}$. We construct a clique $V' \subseteq V$ of size k as follows. Since $C_{i,j}^{\text{succ}} \in \mathcal{I}$ for each $1 \leq i \leq k$ and for each $1 \leq j < k$, we know that $\alpha(x_i^j) = \alpha(x_i^{j'})$, for each $1 \leq i \leq k$ and each $1 \leq j < j' \leq k$. Now, let $V' = \{v_1, \dots, v_k\} \subseteq V$, where $v_i = \alpha(x_i^1)$ for each $1 \leq i \leq k$. We show that V' is a clique. Take arbitrary $v_i, v_j \in V'$ with $1 \leq i < j \leq k$. Since α satisfies $C_{i,j}^E \in \mathcal{I}$, we know that $\{v_i, v_j\} \in E$. In particular, since for each $e \in E$ it holds that $|e| = 2$, we know that $v_i \neq v_j$. Therefore, V' is a k-clique in G. $\square$

COROLLARY 5.4. SMALL-CSP-UNSAT-SUBSET *is* co-W[1]-*hard, even when restricted to CSP instances with degree* 3 *and maximum arity* 2.

Considering all these restrictions simultaneously (bounding the degree, the maximum arity and the domain size of CSP instances) leads to fixed-parameter tractability. In order to show this, we use a bounded search tree algorithm that is essentially the same as the algorithm used to show Proposition 5.1, applied to the setting of CSP instances. For the sake of completeness, we describe it in full.

Again, we describe the algorithm as a non-deterministic algorithm, and then argue that simulating all non-deterministic choices can be done in fixed-parameter linear time. Let $(\mathcal{I}, k)$ be an instance of SMALL-CSP-UNSAT-SUBSET with degree d, maximum arity a, and domain size s. The following procedure decides whether there exists an unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ of size at most k, and computes such a subset if it exists.

The incidence graph of $\mathcal{I}$ has as vertices the set $\text{Var}(\mathcal{I}) \cup \mathcal{I}$, and a variable $x \in \text{Var}(\mathcal{I})$ and a constraint $C \in \mathcal{I}$ are connected with an edge if and only if $x \in \text{Var}(C)$. Without loss of generality, we assume that the incidence graph of $\mathcal{I}$ is connected. Otherwise, we can solve the problem by running the algorithm on each of (the subsets induced by) the connected components.

We guess a constraint $C \in \mathcal{I}$, we let $F_1 := \{C\}$, and we let all variables be unmarked initially. We compute F_{i+1} for $1 \leq i < k$ by means of the following (non-deterministic) rule:

(1) take an unmarked variable $z \in \text{Var}(F_i)$;
(2) guess a non-empty subset $F'_z \subseteq F_z$ for $F_z = \{ C \in \mathcal{I} : z \in \text{Var}(C) \}$;
(3) let $F_{i+1} := F_i \cup F'_z$;
(4) mark z.

If at any point all variables in F_i are marked, we stop computing F_{i+1}. For any F_i, if $|F_i| > k$, we fail. For each F_i, we check whether F_i is unsatisfiable. If it is unsatisfiable, we return with $\mathcal{I}' = F_i$. If it is satisfiable and if it contains no unmarked variables, we fail. Otherwise (that is, if it is satisfiable and contains unmarked variables), we continue with computing F_{i+1}.

It is easy to see that this algorithm is sound. If some $\mathcal{I}' \subseteq \mathcal{I}$ is returned, then $\mathcal{I}'$ is unsatisfiable and $|\mathcal{I}'| \leq k$. In order to see that the algorithm is complete, assume that there exists some unsatisfiable subset $\mathcal{I}' \subseteq \mathcal{I}$ with $|\mathcal{I}'| \leq k$. Then, since we know that the incidence graph of $\mathcal{I}'$ is connected, we know that $\mathcal{I}'$ can be constructed as one of the F_i in the algorithm.

To see that this algorithm witnesses fixed-parameter linearity, we bound its running time. We have to execute the search process at most once for each constraint of $\mathcal{I}$. At each point in the execution of the algorithm, F_i contains at most kda variables, because in each F_{i+1} there are at most da variables more than in the corresponding F_i (at most d constraints are introduced, each with at most a variables). Therefore, there are at most kda choices to take an unmarked variable z. Since each variable occurs in at most d clauses, for each F_z used in the rule, we know $|F_z| \leq d$. Thus, there are at most 2^d possible guesses for F'_z in each execution of the rule. Since we iterate the rule at most k times, we consider at most $(kda2^d)^k$ sets F_i, each containing at most k constraints and containing at most kda variables. Because the domain size is s, we know that each (un)satisfiability check can be done in time $O(s^{kda})$. Therefore, the total running time of the algorithm is $O((kda2^d)^k s^{kda} n)$, where n is the size of the instance.

PROPOSITION 5.5. SMALL-CSP-UNSAT-SUBSET *is fixed-parameter linear when restricted to CSP instances with degree* $f_1(k)$, *with maximum arity* $f_2(k)$, *and domain size* $f_3(k)$, *where* f_1, f_2, f_3 *are computable functions, and* k *is the parameter value.*

PROOF. The result follows directly by using the above algorithm, where we let $d = f_1(k)$, $a = f_2(k)$ and $s = f_3(k)$. □

6. LOCAL BACKBONES

In the introduction, we considered the case study of finding variable assignments that are already forced by a small subset of constraints (we call these *local backbones*). In this section, we consider the parameterized complexity of identifying such local backbones. We consider both the setting of CNF formulas and the setting of CSP instances. In particular, in the setting of CNF formulas, we consider the following parameterized decision problem.

LOCAL-CNF-BACKBONE
Instance: A CNF formula φ, a literal l over some variable $x \in \text{Var}(\varphi)$, and a positive integer k.
Parameter: k.
Question: Is there a subset $\varphi' \subseteq \varphi$ of size k such that $\varphi' \models l$?

For the case of CSP instances, we consider the following parameterized decision problem.

LOCAL-CSP-BACKBONE
Instance: A CSP instance $\mathcal{I}$ over a domain D, a variable $x \in \text{Var}(\mathcal{I})$, a value $d \in D$ and a positive integer k.
Parameter: k.
Question: Is there a subset $\mathcal{I}' \subseteq \mathcal{I}$ of size k such that all solutions α of $\mathcal{I}'$ satisfy that $\alpha(x) = d$?

For the case of CSP instances, we can also consider a dual problem, where the question is whether a small set of constraints already rules out that a variable x is assigned to a given value d (in this case, we speak of *local anti-backbones*). In the case of Boolean domains, this directly reduces to the problem of identifying whether a small set of constraints already enforces that the variable x is assigned to the complementary value $d' \in \{0,1\} \backslash \{d\}$. However, if the domain D possibly has size more than 2, such a direct reduction does not work. Therefore, for the case of CSP instances, we also consider the following parameterized decision problem.

LOCAL-CSP-ANTI-BACKBONE
Instance: A CSP instance $\mathcal{I}$ over a domain D, a variable $x \in \text{Var}(\mathcal{I})$, a value $d \in D$ and a positive integer k.
Parameter: k.
Question: Is there a subset $\mathcal{I}' \subseteq \mathcal{I}$ of size k such that all solutions α of $\mathcal{I}'$ satisfy that $\alpha(x) \neq d$?

In the following, we will show that both in the case of CNF formulas and in the case of CSP instances, the problem of identifying local (anti-)backbones is as hard as identifying small unsatisfiable subsets. In fact, for almost all of the fragments that we considered, the parameterized complexity of both problems is the same. Only for the cases of 0-valid and 1-valid CNF formulas and CSP instances (where the problem of finding unsatisfiable subsets is trivial), the problem of finding local (anti-)backbones is as hard as identifying unsatisfiable subsets in the general case.

6.1. CNF Formulas

Firstly, we show that the problem LOCAL-CNF-BACKBONE is as hard as SMALL-CNF-UNSAT-SUBSET, by providing an fpt-reduction. Moreover, for each class $\mathcal{C}$ of CNF formulas that we considered, applying this reduction to an instance where the formula is in $\mathcal{C}$ results in an instance where the formula is also in $\mathcal{C}$.

LEMMA 6.1. SMALL-CNF-UNSAT-SUBSET *is fpt-reducible to* LOCAL-CNF-BACKBONE.

PROOF. Let (φ, k) be an instance of SMALL-CNF-UNSAT-SUBSET. We construct an instance (ψ, z, k) of LOCAL-CNF-BACKBONE such that $(\varphi, k) \in$ SMALL-CNF-UNSAT-SUBSET if and only if $(\psi, z, k) \in$ LOCAL-CNF-BACKBONE. Take two fresh variables $z, z' \notin \text{Var}(\varphi)$. We then let $\psi = \varphi \cup \{\{\overline{z}, z'\}\}$. We show that φ contains an unsatisfiable subset $\varphi' \subseteq \varphi$ of size k if and only if ψ contains a subset $\psi' \subseteq \psi$ of size k such that $\psi' \models z$.

($\Rightarrow$) Suppose that there is an unsatisfiable subset $\varphi' \subseteq \varphi$ of size k. Then clearly $\varphi' \models z$. Also, since $\varphi \subseteq \psi$, we get that $\varphi' \subseteq \psi$. Thus, $(\psi, z, k) \in$ LOCAL-CNF-BACKBONE.

($\Leftarrow$) Conversely, suppose that there is a subset $\psi' \subseteq \psi$ of size k such that $\psi' \models z$. Since z occurs only negatively in the clause $\{\overline{z}, z'\}$, we know that for $\psi'' = \psi' \backslash \{\{\overline{z}, z'\}\}$ it holds that $\psi'' \models z$. Moreover, $\psi'' \subseteq \varphi$ and $|\psi''| \leq k$. Then, since φ does not contain any occurrence of the variable z, we know that ψ'' is unsatisfiable. Thus $(\varphi, k) \in$ SMALL-CNF-UNSAT-SUBSET. □

Next, we show that we can also construct an fpt-reduction in the other direction. For each class $\mathcal{C}$ of CNF formulas that is closed under instantiation, applying this reduction to an instance where the formula is in $\mathcal{C}$ results in an instance where the formula is also in $\mathcal{C}$. This is the case for all classes of CNF formulas that we considered, except for the cases of 0-valid and 1-valid CNF formulas.

LEMMA 6.2. LOCAL-CNF-BACKBONE *is fpt-reducible to* SMALL-CNF-UNSAT-SUBSET.

PROOF. Let (φ, l, k) be an instance of LOCAL-CNF-BACKBONE. We construct an instance (ψ, k) of SMALL-CNF-UNSAT-SUBSET. We let $\psi = \varphi|_{\overline{l}}$.

We may assume without loss of generality that the clause $\{l\}$ does not appear in φ (if this is the case, the instance is a trivial yes-instance for every $k \geq 1$). Therefore, ψ does not contain the empty clause.

We claim that $(\varphi, l, k) \in$ LOCAL-CNF-BACKBONE if and only if $(\psi, k) \in$ SMALL-CNF-UNSAT-SUBSET.

$(\Rightarrow)$ Assume that there is a subset $\varphi' \subseteq \varphi$ of size k such that $\varphi' \models l$. Now consider the set $\psi' = \varphi'|_{\overline{l}}$. Clearly, $|\psi'| \leq k$, and $\psi' \subseteq \psi$. Also, since $\varphi' \models l$, we know that ψ' is unsatisfiable. Therefore, $(\psi, k) \in$ SMALL-CNF-UNSAT-SUBSET.

$(\Leftarrow)$ Conversely, suppose that there is an unsatisfiable subset $\psi' \subseteq \psi$ of size k. By construction of ψ, we know that then there must be a subset $\varphi' \subseteq \varphi$ such that $\psi' = \varphi'|_{\overline{l}}$. Moreover, $|\varphi'| = k$. To show that $\varphi' \models l$, suppose that there is a truth assignment α that satisfies φ' and for which $\alpha(l) = 0$. Then, α would also satisfy ψ'. This is a contradiction with our assumption that ψ' is unsatisfiable, and therefore we can conclude that $\varphi' \models l$. Thus, $(\varphi, l, k) \in$ LOCAL-CNF-BACKBONE. □

Lemmas 6.1 and 6.2 directly give us the following result about the relation between the parameterized complexity of the problems SMALL-CNF-UNSAT-SUBSET and LOCAL-CNF-BACKBONE.

PROPOSITION 6.3. *For all classes of CNF formulas that we considered, with the exception of 0-valid and 1-valid CNF formulas, the problems* SMALL-CNF-UNSAT-SUBSET *and* LOCAL-CNF-BACKBONE *have the same parameterized complexity (modulo fpt-reductions).*

In addition, the proof of Lemma 6.1 can straightforwardly be modified to work also for the case of affine formulas (by replacing the new clause $\{\overline{z}, z'\}$ by the affine clause $(z \oplus z')$, for instance). The proof of Lemma 6.2 also works for the case of affine formulas. This gives us the following result.

PROPOSITION 6.4. *The problems* SMALL-CNF-UNSAT-SUBSET *and* LOCAL-CNF-BACKBONE *have the same parameterized complexity (modulo fpt-reductions) also for the case where the input formulas are affine. In particular, in this setting, the problem* LOCAL-CNF-BACKBONE *is* W[1]*-hard.*

For the case of 0-valid and 1-valid CNF formulas, the problem LOCAL-CNF-BACKBONE turns out to be harder than SMALL-CNF-UNSAT-SUBSET.

LEMMA 6.5. LOCAL-CNF-BACKBONE *is* W[1]*-hard when restricted to 0-valid CNF formulas.*

PROOF. We show this by means of a reduction from SMALL-CNF-UNSAT-SUBSET. Let (φ, k) be an instance of SMALL-CNF-UNSAT-SUBSET. We construct an instance $(\psi, \overline{z}, k)$ of LOCAL-CNF-BACKBONE, where ψ is 0-valid. Take a fresh variable $z \notin \text{Var}(\varphi)$. Then, let $\psi = \{\, c \cup \{\overline{z}\} : c \in \varphi \,\}$. Clearly, ψ is 0-valid, because each clause of ψ contains the negative literal $\overline{z}$. We show that $(\varphi, k) \in$ SMALL-CNF-UNSAT-SUBSET if and only if $(\psi, \overline{z}, k) \in$ LOCAL-CNF-BACKBONE.

$(\Rightarrow)$ Suppose there exists an unsatisfiable subset $\varphi' \subseteq \varphi$ of size k. Consider the set $\psi' = \{\, c \cup \{\overline{z}\} : c \in \varphi' \,\}$. Clearly, $\psi' \subseteq \psi$ and $|\psi'| = k$. To show that $\psi' \models \overline{z}$, suppose that there exists a truth assignment $\alpha : \text{Var}(\psi) \to \{0, 1\}$ with $\alpha(z) = 1$ that satisfies ψ'. It is straightforward to verify that α then satisfies φ', which is a contradiction with our assumption that φ' is unsatisfiable. Therefore, we can conclude that $\psi' \models \overline{z}$, and thus $(\psi, \overline{z}, k) \in$ LOCAL-CNF-BACKBONE.

$(\Leftarrow)$ Conversely, assume that there exists a subset $\psi' \subseteq \psi$ of size k such that $\psi' \models \overline{z}$. Then consider the set $\varphi' = \{\, c \backslash \{\overline{z}\} : c \in \psi' \,\}$. Clearly, $\varphi' \subseteq \varphi$ and $|\varphi'| = k$. To show that φ' is unsatisfiable, suppose that there exists a truth assignment $\alpha : \text{Var}(\varphi) \to \{0, 1\}$ that satisfies φ'. Consider the truth assignment $\alpha' : \text{Var}(\psi) \to \{0, 1\}$ that is defined by letting $\alpha'(z) = 1$ and letting $\alpha'(x) = \alpha(x)$ for all $x \in \text{Var}(\varphi)$. Then α' simultaneously satisfies ψ' and z, which is a contradiction with our assumption that $\psi' \models \overline{z}$. Therefore, we can conclude that φ' is unsatisfiable, and thus that $(\varphi, k) \in$ SMALL-CNF-UNSAT-SUBSET. □

By Lemmas 6.2 and 6.5 (and by the straightforward extension of the latter to the case of 1-valid CNF formulas), we then get the following result.

PROPOSITION 6.6. LOCAL-CNF-BACKBONE *is* W[1]*-complete when restricted to 0-valid formulas or when restricted to 1-valid formulas.*

6.2. CSP Instances

Next, we turn our attention to the case of CSP instances. We firstly show by means of an fpt-reduction that the problems LOCAL-CSP-BACKBONE and LOCAL-CSP-ANTI-BACKBONE are at least as hard as the problem SMALL-CSP-UNSAT-SUBSET. This reduction has the property that for each class $\mathcal{C}$ of CSP instances that we considered, applying this reduction to an instance where the CSP instance is in $\mathcal{C}$ results in an instance where the CSP instance is also in $\mathcal{C}$.

LEMMA 6.7. *The problem* SMALL-CSP-UNSAT-SUBSET *is fpt-reducible to the problems* LOCAL-CSP-BACKBONE *and* LOCAL-CSP-ANTI-BACKBONE.

PROOF. We first give an fpt-reduction to LOCAL-CSP-BACKBONE. The main idea behind this reduction is similar to the idea of the proof of Lemma 6.1. Namely, if a CSP instance $\mathcal{I}$ has a small unsatisfiable subset, then any variable of $\mathcal{I}$ is a local backbone. Let $(\mathcal{I}, k)$ be an instance of SMALL-CSP-UNSAT-SUBSET. We construct an equivalent instance $\mathcal{I}'$ of LOCAL-CSP-BACKBONE by introducing an additional fresh variable z to $\mathcal{I}$ (for instance, by adding an additional constraint C with $\text{Var}(C) = \{z\}$). We may assume that the variable z is unconstrained and can get any value $d \in D$ in any solution. Since there is no constraint in $\mathcal{I}'$ that directly enforces variable v to take any particular value, the only possibility for v to be a backbone is if $\mathcal{I}'$ has no solutions, and thus is unsatisfiable. We then know that $\mathcal{I}$ contains an unsatisfiable subset of at most k constraints if and only if $\mathcal{I}'$ contains an unsatisfiable subset of at most k constraints, which holds if and only if some subset of $\mathcal{I}$ of at most k constraints forces each solution α to satisfy $\alpha(z) = 0$.

This reductions is also an fpt-reduction to LOCAL-CSP-ANTI-BACKBONE, since $\mathcal{I}$ contains an unsatisfiable subset containing at most k constraints if and only if some subset of $\mathcal{I}'$ containing at most k constraints forces each solution α to satisfy $\alpha(z) \neq d_0$, for any $d_0 \in D$. □

Next, we show that we can also construct an fpt-reduction in the other direction. For each class $\mathcal{C}$ of CSP instances that is closed under partial assignment, applying this reduction to an instance where the CSP instance is in $\mathcal{C}$ results in an instance where the CSP instance is also in $\mathcal{C}$. This is the case for all classes of CSP instances that we considered, except for the cases of 0-valid and 1-valid CSP instances.

LEMMA 6.8. *The problems* LOCAL-CSP-BACKBONE *and* LOCAL-CSP-ANTI-BACKBONE *are fpt-reducible to* SMALL-CSP-UNSAT-SUBSET.

PROOF. We firstly give an fpt-reduction for the case of LOCAL-CSP-BACKBONE. Let $(\mathcal{I}, x, d, k)$ be an instance of LOCAL-CSP-BACKBONE. We construct an equivalent instance $(\mathcal{I}', k)$ of SMALL-CSP-UNSAT-SUBSET as follows. For each constraint $C = (S, R) \in \mathcal{I}$, we add the constraint $C' = (S, R')$ to $\mathcal{I}'$, whose scope S is identical to the scope of C, but whose constraint relation R' differs from the constraint relation R of C as follows. The constraint relation R' contains exactly those tuples from R that do not set the variable x to d. So, if $x \notin \text{Var}(C)$, then $R' = R$. We show that $(\mathcal{I}', k)$ is a yes-instance of SMALL-CSP-UNSAT-SUBSET if and only if $(\mathcal{I}, x, d, k)$ is a yes-instance of LOCAL-CSP-BACKBONE.

$(\Rightarrow)$ Suppose that there is an unsatisfiable subset $\mathcal{J}' \subseteq \mathcal{I}'$ of size k. Consider the subset $\mathcal{J} \subseteq \mathcal{I}$ that corresponds to $\mathcal{J}'$, that is, $\mathcal{J}$ consists of those constraints C for which $C' \in \mathcal{J}'$. Clearly, $|\mathcal{J}| \leq k$. We show that every solution α of $\mathcal{J}$ satisfies $\alpha(x) = d$. Take an arbitrary solution α of $\mathcal{J}$, and to derive a contradiction, suppose that $\alpha(x) \neq d$. Now, by construction of the constraints C' in $\mathcal{I}'$, we know that α satisfies $\mathcal{J}'$, which is a contradiction with our assumption that $\mathcal{J}'$ is unsatisfiable. Therefore, we can conclude that every solution α of $\mathcal{J}$ satisfies that $\alpha(x) = d$. Thus, $(\mathcal{I}, x, d, k) \in$ LOCAL-CSP-BACKBONE.

($\Leftarrow$) Conversely, suppose that there is a subset $\mathcal{J} \subseteq \mathcal{I}$ of size k such that all solutions α of $\mathcal{J}$ satisfy that $\alpha(x) = d$. Then, consider the subset $\mathcal{J}' \subseteq \mathcal{I}'$ consisting of exactly those constraints C' for which $C \in \mathcal{J}$. Clearly, $|\mathcal{J}'| \leq k$. We show that $\mathcal{J}'$ is unsatisfiable. There are two possibilities: either (1) $x \notin \text{Var}(\mathcal{J}')$ or (2) $x \in \text{Var}(\mathcal{J}')$. In case (1), we know that $\mathcal{J} = \mathcal{J}'$, and moreover we know that $\mathcal{J}'$ is unsatisfiable. Consider case (2). To derive a contradiction, suppose that there is some solution α of $\mathcal{J}'$. Then, by construction of the constraints in $\mathcal{I}'$, we know that $\alpha(x) \neq d$. Moreover, then α is also a solution of $\mathcal{J}$. This is a contradiction with our assumption that all solutions of $\mathcal{J}$ set x to d. Therefore, we can conclude that $\mathcal{J}'$ is unsatisfiable. Thus, $(\mathcal{I}', k) \in$ SMALL-CSP-UNSAT-SUBSET.

A similar fpt-reduction works for the case of LOCAL-CSP-ANTI-BACKBONE, with the difference that for the constraints C' in $\mathcal{I}'$ the constraint relation R' contains exactly those tuples of the constraint relation R of the original constraint C that do not set x to any value $d' \in D \backslash \{d\}$. $\square$

Lemmas 6.7 and 6.8 give us the following result about the relation between the parameterized complexity of the problems SMALL-CSP-UNSAT-SUBSET, LOCAL-CSP-BACKBONE, and LOCAL-CSP-ANTI-BACKBONE.

PROPOSITION 6.9. *For all classes of CSP instances that we considered, with the exception of 0-valid and 1-valid CSP instances, the problems* SMALL-CNF-UNSAT-SUBSET, LOCAL-CNF-BACKBONE, *and* LOCAL-CSP-ANTI-BACKBONE *have the same parameterized complexity (modulo fpt-reductions).*

For the case of 0-valid and 1-valid CSP instances, it turns out that the problems LOCAL-CSP-BACKBONE and LOCAL-CSP-ANTI-BACKBONE are in fact A[2]-complete. This is in contrast with the (trivial) polynomial-time solvability of SMALL-CSP-UNSAT-SUBSET with the same restrictions. We prove the hardness result for the restriction to 0-valid instances. The result for 1-valid instances then follows by a symmetry argument.

LEMMA 6.10. LOCAL-CSP-BACKBONE *is* A[2]-*hard when restricted to Boolean CSP instances that are* 0-*valid.*

PROOF PROOF (IDEA). Similarly to the proof of Lemma 6.5, we can construct an fpt-reduction from the problem SMALL-CSP-UNSAT-SUBSET restricted to Boolean CSP instances, to the problem LOCAL-CSP-BACKBONE restricted to 0-valid Boolean CSP instances. $\square$

By Lemmas 6.8 and 6.10 (and by the straightforward extension of the latter to the case of 1-valid CSP instances), we then get the following result.

PROPOSITION 6.11. LOCAL-CSP-BACKBONE *and* LOCAL-CSP-ANTI-BACKBONE *are* A[2]-*complete, when restricted to Boolean CSP instances that are* 0-*valid or when restricted to Boolean CSP instances that are* 1-*valid.*

6.3. Iterative Local Backbones

We showed that the problem of identifying local backbones is as hard as the problem of finding small unsatisfiable subsets (if not harder), for all classes of CNF formulas and CSP instances that we considered. In the introduction, we briefly discussed the notion of iterative local backbones, which are local backbones that are obtained by repeatedly finding local backbones and its corresponding (truth) value and instantiating them. In this section, we briefly consider the parameterized complexity of identifying iterative local backbones.

Firstly, we show that fixed-parameter tractability results for finding local backbones (for classes of CNF instances that are closed under instantiation) carry over to the setting of iterative local backbones.

PROPOSITION 6.12. *Let $\mathcal{C}$ be a class of CNF formulas that is closed under variable instantiation. If the problem* LOCAL-CNF-BACKBONE *is fixed-parameter tractable when restricted to CNF formulas in $\mathcal{C}$, then finding all iterative local backbones of order at most k (and their corresponding truth value) can be found in fixed-parameter tractable time.*

PROOF. The fixed-parameter tractable algorithm A to solve LOCAL-CNF-BACKBONE can be applied a polynomial number of times to find all iterative local backbones of order at most k. Let $\varphi \in \mathcal{C}$ be a CNF formula, and let $k \in \mathbb{N}$. We describe an algorithm to find all iterative local backbones of order at most k. The algorithm works in rounds. In each round, we use the algorithm A to decide if $(\varphi, l, k) \in$ LOCAL-CNF-BACKBONE, for each literal l over the variables in Var(φ). Each literal l for which $(\varphi, l, k) \in$ LOCAL-CNF-BACKBONE, we add to the collection of found iterative local backbones of order at most k. Moreover, we update φ by instantiating l. It is straightforward to verify that all the implied literals that this algorithm identifies are in fact iterative local backbones of order at most k, and that any iterative local backbone of order at most k is identified by the algorithm in at most $|\text{Var}(\varphi)|$ rounds. Moreover, the algorithm runs in fixed-parameter tractable time (with respect to the parameter k). $\square$

For the case of CSP instances, the notion of iterative local backbones can be defined analogously, and in this setting we can get a similar result.

PROPOSITION 6.13. *Let $\mathcal{C}$ be a class of CSP instances that is closed under partial assignment. If the problem* LOCAL-CSP-BACKBONE *is fixed-parameter tractable when restricted to CSP instances in $\mathcal{C}$, then finding all iterative local backbones of order at most k (and their corresponding value) can be found in fixed-parameter tractable time.*

PROOF. The proof is entirely analogous to the proof of Proposition 6.12. $\square$

For essentially all classes of CNF formulas and (Boolean) CSP instances that we considered in Sections 4 and 5 for which the problems SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET are fixed-parameter intractable (e.g., W[1]-hard), these intractability results can be extended to the problem of finding iterative local backbones (of order at most k). Let $\mathcal{C}$ be a class of CNF formulas or CSP instances. By the proof of Lemmas 6.1 and 6.7, we know that if the problem of identifying small unsatisfiable subsets restricted to $\mathcal{C}$ is shown to be W[1]-hard by means of an fpt-reduction that does not involve instances containing backbones of order at most k, then the problem of identifying iterative local backbones of order at most k is W[1]-hard as well. The hardness proofs for SMALL-CNF-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET that we presented in Sections 4 and 5 have this property (or can be modified straightforwardly in such a way that they have this property)—with the exception of Proposition 4.4.

Interestingly, there is one class of CNF formulas for which the problem of identifying iterative local backbones is easier than identifying local backbones. This is the class of definite Horn formulas. We can use the W[1]-hardness result of Lemma 4.3 to show W[1]-hardness for the problem LOCAL-CNF-BACKBONE restricted to definite Horn formulas.

PROPOSITION 6.14. LOCAL-CNF-BACKBONE *is* W[1]*-hard even when restricted to CNF formulas that are definite Horn, and that contain only clauses of size at most* 3*.*

PROOF. We describe an fpt-reduction from SHORT-HYPERPATH. Let (φ, s, t, k) be an instance of SHORT-HYPERPATH. By Lemma 4.3, we may assume without loss of generality that φ contains only definite Horn clauses of size at most 3. Then let $\varphi' = \varphi \cup \{\{s\}\}$. We have that $(\varphi, s, t, k) \in$ SHORT-HYPERPATH if and only if there is a subset $\varphi'' \subseteq \varphi'$ of size at most $k' = k + 1$ such that $\varphi'' \models \neg t$. $\square$

In contrast, for definite Horn formulas, finding iterative local backbones of order at most k can be done in polynomial time, for any $k \in \mathbb{N}$. We show that for any k and any definite Horn formula φ, the set of iterative local backbones of φ of order at most k coincides with the set of backbones of φ. Since for definite Horn formulas, deciding whether a literal l is entailed can be done in linear time [Dowling and Gallier 1984], computing the set of iterative local backbones of a definite Horn formula φ of order at most k can be done in polynomial time.

PROPOSITION 6.15. *Let φ be a definite Horn formula, and let $x \in \text{Var}(\varphi)$ be a variable occurring in φ. Then x is an iterative local backbone of φ of order 1 if and only if $\varphi \models x$.*

PROOF. One direction follows immediately: if x is an iterative local backbone of φ of order 1, then x is a backbone of φ. Since definite Horn formulas can only entail positive literals (the all-ones truth assignment satisfies any definite Horn formula), we then know that $\varphi \models x$.

Conversely, assume that $\varphi \models x$. It is well-known that the method of forward chaining (that is, applying the rule of modus ponens to derive the entailed unit clause $\{x\}$ from a definite Horn clause $\{\neg x_1, \dots, \neg x_m, x\}$ and previously derived entailed unit clauses $\{x_1\}, \dots, \{x_m\}$) is a complete method for deriving entailed unit clauses for definite Horn formulas. Therefore, there must be a forward chaining derivation witnessing that $\varphi \models x$. It is now straightforward to show by induction on the structure of this derivation that x is an iterative local backbone of φ of order 1. □

6.3.1. Relation to generalized unit-refutation completeness. Somewhat related to the method of computing enforced assignments via iterative local backbones is the mechanism used to define *unit-refutation complete formulas of level k* [Gwynne and Kullmann 2013; Kullmann 1999]. This mechanism is based on mappings r_k from CNF formulas to CNF formulas. For a nonnegative integer k, the mapping r_k is defined inductively as follows. In the case for $k = 0$, the definition states that $r_0(\varphi) = \{\bot\}$ if $\bot \in \varphi$, and $r_0(\varphi) = \varphi$ otherwise. In the case for $k > 0$, the definition states that $r_k(\varphi) = r_k(\varphi|_l)$ if there exists a literal $l \in \text{Lit}(\varphi)$ such that $r_{k-1}(\varphi|_{\bar{l}}) = \{\bot\}$, and $r_k(\varphi) = \varphi$ otherwise. In particular, the mapping r_1 computes the result of applying unit propagation. Note that the result of $r_k(\varphi)$ is the application of a number of forced assignments to φ, i.e., $r_k(\varphi) = \varphi|_L$ for some $L \subseteq \text{Lit}(\varphi)$ such that for all $l \in L$ it holds that $\varphi \models l$. We let $L_k^{\text{UC}}(\varphi)$ denote the set of forced literals that are computed by r_k, i.e., $L_k^{\text{UC}}(\varphi) = L \subseteq \text{Lit}(\varphi)$ such that $r_k(\varphi) = \varphi|_L$. Similarly, we let $L_k^{\text{ILB}}(\varphi)$ denote the set of forced literals that are found by computing iterative local backbones of order at most k.

The following observations relate the two mechanisms. Let φ be an arbitrary CNF formula. We have that $L_1^{\text{UC}}(\varphi) = L_1^{\text{ILB}}(\varphi)$. In fact, this set contains exactly those enforced literals that can be found by unit propagation. Also, for any $k \geq 2$ we have that $L_k^{\text{ILB}}(\varphi) \subsetneq L_k^{\text{UC}}(\varphi)$. The inclusion follows from the fact that each minimal subset φ' of size at most k that enforces a literal l has at most k literals (which is a direct result of Tarsi's Lemma). Whenever l is identified as an enforced literal in the computation of iterative local backbones of order at most k, it can then also be computed by r_k by first guessing $\bar{l}$, and subsequently obtaining a contradiction for each instantiation of the other variables in $\text{Var}(\varphi')$. In order to see that the inclusion is strict, consider the family of formulas $(\varphi_n)_{n \in \mathbb{N}}$, where $\varphi_n = \{\, \{\neg x_i, x_{i+1}\} : 1 \leq i < n \,\} \cup \{\neg x_n, \neg x_1\}$. For each φ_n, we know that $\varphi_n \models \neg x_1$. Furthermore, we have that $\neg x_1 \in L_2^{\text{UC}}(\varphi_n)$, but x_1 is not an iterative local backbone of φ_n of any order $k < n$.

7. CONCLUSION

We studied the problem of identifying whether a given set of constraints has a small unsatisfiable subset (and if it does, finding such a subset) from a parameterized complexity point of view. We studied both the case where the set of constraints is given in the form of a CNF (or affine) formula, and the case where the set of constraints is given in the form of a CSP instance. In the general case of the problem, we showed that the problem is harder in the case of CSP instances (we showed that in this case the problem is A[2]-complete) than in the case of CNF instances (in this case the problem was already known to be W[1]-complete). Interestingly, this is one of the first problems to be shown complete for the parameterized intractability class A[2].

Then, we considered various restricted classes of instances over a Boolean domain. In particular, we considered the classes of instances (propositional formulas and CSP instances) induced by the constraint languages identified by Schaefer as the maximal constraint languages with a tractable satisfiability problem. In the case of propositional formulas, we showed that the problem remains W[1]-hard in all (non-trivial) cases, except for Krom (2CNF) formulas. In the case of CSP instances, we showed that the problem is at least as hard as the corresponding problem for propositional formulas (and in many cases even harder), and that in all cases the problem is fixed-parameter intractable (that is, at least W[1]-hard).

Both for the setting of CNF formulas and for the setting of CSP instances we also identified fragments for which the problem is fixed-parameter tractable. For CNF formulas, any class of instances where the degree (the maximum number of times that any variable occurs in the instance) is bounded by a function of the parameter leads to fixed-parameter tractability. For CSP instances, one needs to bound the maximum arity of constraints and the domain size (by a function of the parameter) in addition to the degree, to obtain fixed-parameter tractability.

Finally, we related the problem of finding small unsatisfiable subsets to the problem of identifying whether a small subset of constraints already enforces or rules out a variable-value assignment. We showed that the latter problem is just as hard as the former problem for all fragments that we considered, with one exception. For the class of 0-valid and 1-valid constraints, the problem of finding small unsatisfiable subsets is trivial (there are no unsatisfiable subsets), whereas the problem of finding variable-value assignments that are enforced by a small subset is W[1]-complete for CNF formulas and A[2]-complete for CSP instances.

Future research includes extending this parameterized complexity analysis to further formalisms that express constraints. Natural candidates of such formalisms can be found, for instance, in the area of Satisfiability Modulo Theories (SMT). There, sets of constraints are usually expressed as CNF formulas where the propositional variables are atomic statements over an underlying theory (e.g., linear arithmetic).

REFERENCES

Faisal N. Abu-Khzam, Michael A. Langston, Pushkar Shanbhag, and Christopher T. Symons. 2006. Scalable Parallel Algorithms for FPT Problems. *Algorithmica* 45, 3 (2006), 269–284.

Ron Aharoni and Nathan Linial. 1986. Minimal Non-Two-Colorable Hypergraphs and Minimal Unsatisfiable Formulas. *J. Combin. Theory Ser. A* 43 (1986), 196–204.

Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity – A Modern Approach*. Cambridge University Press. I–XXIV, 1–579 pages.

Fahiem Bacchus and George Katsirelos. 2015. Using minimal correction sets to more efficiently compute minimal unsatisfiable sets. In *Proceedings of the 27th International Conference on Computer Aided Verification (CAV 2015) (Lecture Notes in Computer Science)*, Vol. 9207. Springer Verlag, 70–86.

Anton Belov, Inês Lynce, and João Marques-Silva. 2012. Towards efficient MUS extraction. *AI Commun.* 25, 2 (2012), 97–116.

Anton Belov and João Marques-Silva. 2012. MUSer2: An Efficient MUS Extractor. *J on Satisfiability, Boolean Modeling and Computation* 8, 1/2 (2012), 123–128.

Joshua Buresh-Oppenheim and David G. Mitchell. 2006. Minimum Witnesses for Unsatisfiable 2CNFs. In *Proceedings of SAT 2006, Ninth International Conference on Theory and Applications of Satisfiability Testing, August 12-15, 2006, Seattle, WA, USA (Lecture Notes in Computer Science)*, Armin Biere and Carla P. Gomes (Eds.), Vol. 4121. 42–47.

James Cheetham, Frank Dehne, Andrew Rau-Chaplin, Ulrike Stege, and Peter J. Taillon. 2003. Solving large FPT problems on coarse-grained parallel machines. *J. of Computer and System Sciences* 67, 4 (2003), 691–706.

Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David Juedes, Iyad Kanj, and Ge Xia. 2005. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation* 201, 2 (2005), 216–231.

Jianer Chen, Xiuzhen Huang, Iyad Kanj, and Ge Xia. 2006. Strong computational lower bounds via parameterized complexity. *J. of Computer and System Sciences* 72, 8 (2006), 1346–1367.

Jianer Chen and Iyad Kanj. 2012. Parameterized Complexity and Subexponential-Time Computability. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*

(Lecture Notes in Computer Science), Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx (Eds.), Vol. 7370. Springer Verlag, 162–195.

Alessandro Cimatti, Alberto Griggio, and Roberto Sebastiani. 2011. Computing Small Unsatisfiable Cores in Satisfiability Modulo Theories. *J. Artif. Intell. Res.* 40 (2011), 701–728.

Jason Crampton, Gregory Gutin, and Daniel Karapetyan. 2015. Valued workflow satisfiability problem. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies (SACMAT 2015)*. Assoc. Comput. Mach., New York, 3–13.

Marek Cygan, Fedor V Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

Adnan Darwiche and Pierre Marquis. 2002. A Knowledge Compilation Map. *J. Artif. Intell. Res.* 17 (2002), 229–264.

William F. Dowling and Jean H. Gallier. 1984. Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulae. *J. Logic Programming* 1, 3 (1984), 267–284.

Rodney G. Downey and Michael R. Fellows. 1995a. Fixed-parameter tractability and completeness I: Basic results. *SIAM J. Comput.* 24, 4 (1995), 873–921.

Rodney G. Downey and Michael R. Fellows. 1995b. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science* 141, 1-2 (1995), 109–131.

Rodney G. Downey and Michael R. Fellows. 1999. *Parameterized Complexity*. Springer Verlag, New York.

Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer Verlag.

Olivier Dubois and Gilles Dequen. 2001. A backbone-search heuristic for efficient solving of hard 3-SAT formulae. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001*, Bernhard Nebel (Ed.). 248–253.

Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. 2009. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science* 410, 1 (2009), 53–61.

Michael R. Fellows, Stefan Szeider, and Graham Wrightson. 2006. On Finding Short Resolution Refutations and Small Unsatisfiable Subsets. *Theoretical Computer Science* 351, 3 (2006), 351–359.

Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series, Vol. XIV. Springer Verlag, Berlin.

Giorgio Gallo, Giustino Longo, Stefano Pallotino, and Sang Nguyen. 1993. Directed hypergraphs and applications. *Discrete Applied Mathematics* 42 (1993), 177–201.

Serge Gaspers and Stefan Szeider. 2011. The Parameterized Complexity of Local Consistency. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP 2011) (Lecture Notes in Computer Science)*, Jimmy Ho-Man Lee (Ed.), Vol. 6876. Springer Verlag, 302–316.

Georg Gottlob and Stefan Szeider. 2006. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction, and database problems. *Comput. J.* 51, 3 (2006), 303–325.

Matthew Gwynne and Oliver Kullmann. 2013. Generalising and Unifying SLUR and Unit-Refutation Completeness.. In *SOFSEM 2013: Theory and Practice of Computer Science (Lecture Notes in Computer Science)*, Vol. 7741. 220–232.

Ronald de Haan, Iyad Kanj, and Stefan Szeider. 2013a. Local Backbones. In *Proceedings of SAT 2013, 16th International Conference on Theory and Applications of Satisfiability Testing, July 8-12, 2013, Helsinki, Finland (Lecture Notes in Computer Science)*, Matti Järvisalo and Allen Van Gelder (Eds.), Vol. 7962. Springer Verlag, 377–393.

Ronald de Haan, Iyad Kanj, and Stefan Szeider. 2013b. *Local Backbones*. Technical Report 1304.5479. arXiv.org. (updated version of [de Haan et al. 2013a]).

Ronald de Haan, Iyad Kanj, and Stefan Szeider. 2014. Small Unsatisfiable Subsets in Constraint Satisfaction. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus, November 10-12, 2014*. 429–436. DOI: http://dx.doi.org/10.1109/ICTAI.2014.72

Fred Hemery, Christophe Lecoutre, Lakhdar Sais, and Frédéric Boussemart. 2006. Extracting MUCs from constraint networks. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006) (Frontiers in Artificial Intelligence and Applications)*, Vol. 141. 113–117.

Timon Hertli, Robin A. Moser, and Dominik Scheder. 2011. Improving PPSZ for 3-SAT using Critical Variables.. In *Symposium on Theoretical Aspects of Computer Science*, Thomas Schwentick and Christoph Dürr (Eds.), Vol. 9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 237–248.

Holger H. Hoos and Thomas Stützle. 2000. SATLIB: An Online Resource for Research on SAT. In *SAT2000: Highlights of Satisfiability Research in the year 2000*, Ian Gent, Hans van Maaren, and Toby Walsh (Eds.). Kluwer Academic, 283–292.

Falk Hüffner, Sebastian Wernicke, and Thomas Zichner. 2008. Algorithm engineering for color-coding with applications to signaling pathway detection. *Algorithmica* 52, 2 (2008), 114–132.

Alexey Ignatiev, Alessandro Previti, Mark Liffiton, and Joao Marques-Silva. 2015. Smallest MUS extraction with minimal hitting set dualization. In *Proceedings of the 21st International Conference on Principles and Practice of Constraint Programming (CP 2015) (Lecture Notes in Computer Science)*, Vol. 9255. Springer Verlag, 173–182.

Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which problems have strongly exponential complexity? *J. of Computer and System Sciences* 63, 4 (2001), 512–530.

Henry Kautz and Bart Selman. 1996. Pushing the envelope: planning, propositional logic, and stochastic search. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence (AAAI'96)*. AAAI Press, 1194–1201.

Philip Kilby, John K. Slaney, Sylvie Thiébaux, and Toby Walsh. 2005. Backbones and Backdoors in Satisfiability.. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA (AAAI 2005)*. 1368–1373.

Oliver Kullmann. 1999. *Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs*. Technical Report TR99–041. *Electronic Colloquium on Computational Complexity* (ECCC).

Oliver Kullmann. 2000a. An application of matroid theory to the SAT problem. In *Fifteenth Annual IEEE Conference on Computational Complexity*. 116–124. See also TR00-018, Electronic Colloquium on Computational Complexity (ECCC), March 2000.

Oliver Kullmann. 2000b. *An application of matroid theory to the SAT problem*. Technical Report TR00–018. *Electronic Colloquium on Computational Complexity* (ECCC).

Michael A. Langston, Andy D. Perkins, Arnold M. Saxton, Jon A. Scharff, and Brynn H. Voy. 2008. Innovative computational methods for transcriptomic data analysis: A case study in the use of FPT for practical algorithm design and implementation. *Comput. J.* 51, 1 (2008), 26–38.

Paolo Liberatore. 2005. Redundancy in logic I: CNF propositional formulae. *Artificial Intelligence* 163, 2 (2005), 203–232.

Inês Lynce and João P. Marques Silva. 2004. On Computing Minimum Unsatisfiable Cores. In *Proceedings of SAT 2004 (Seventh International Conference on Theory and Applications of Satisfiability Testing, 10–13 May, 2004, Vancouver, BC, Canada)*.

João Marques-Silva. 2012. Computing Minimally Unsatisfiable Subformulas: State of the Art and Future Directions. *Journal of Multiple-Valued Logic and Soft Computing* (2012), 163–183.

Albert R. Meyer and Larry J. Stockmeyer. 1972. The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space. In *SWAT*. IEEE Computer Soc., 125–129.

Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, Oxford.

Christos H. Papadimitriou. 1994. *Computational Complexity*. Addison-Wesley.

Christos H. Papadimitriou and Mihalis Yannakakis. 1999. On the complexity of database queries. *J. of Computer and System Sciences* 58, 3 (1999), 407–427.

Andrew J. Parkes. 1997. Clustering at the phase transition. In *14th National Conference on Artificial Intelligence (AAAI'97)*. AAAI Press, 340–345.

Daniele Prelotani. 1996. Efficiency and Stability of Hypergraph SAT Algorithms. In *Cliques, Coloring and Satisfiability*, David S. Johnson and Michael A. Trick (Eds.). AMS, 479–498.

Chantal Roth-Korostensky. 2000. *Algorithms for Building Multiple Sequence Alignments and Evolutionary Trees*. Ph.D. Dissertation. 13550, ETH Zürich.

Marko Samer and Stefan Szeider. 2010. Constraint satisfaction with bounded treewidth revisited. *J. of Computer and System Sciences* 76, 2 (2010), 103–114.

Thomas J. Schaefer. 1978. The complexity of satisfiability problems. In *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978)*. ACM, 216–226.

Johannes Schneider, Christine Froschhammer, Ingo Morgenstern, Thomas Husslein, and Johannes Maria Singer. 1996. Searching for backbones – an efficient parallel algorithm for the traveling salesman problem. *Computer Physics Communications* 96 (1996), 173–188.

John K. Slaney and Toby Walsh. 2001. Backbones in Optimization and Approximation. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001*, Bernhard Nebel (Ed.). 254–259.

Yinglei Song, Chunmei Liu, Xiuzhen Huang, Russell L. Malmberg, Ying Xu, and Liming Cai. 2006. Efficient Parameterized Algorithms for Biopolymer Structure-Sequence Alignment. *IEEE/ACM Trans. Comput. Biology Bioinform.* 3, 4 (2006), 423–432.

Ulrike Stege. 2000. *Resolving Conflicts from Problems in Computational Biology*. Ph.D. Dissertation. 13364, ETH Zürich.

Larry J. Stockmeyer. 1976. The polynomial-time hierarchy. *Theoretical Computer Science* 3, 1 (1976), 1–22.

Ofer Strichman. 2000. Tuning SAT Checkers for Bounded Model Checking. In *Computer Aided Verification: 12th International Conference (CAV 2000)*. Springer-Verlag, London, UK, 480–494.

Celia Wrathall. 1976. Complete Sets and the Polynomial-Time Hierarchy. *Theoretical Computer Science* 3, 1 (1976), 23–33.

Received January 2017; revised January 2017; accepted January 2017

A. EMPIRICAL RESULTS: (ITERATIVE) LOCAL BACKBONES

In this appendix, we provide some empirical results that show the low (iterative) order of many backbones in several SAT instances from various domains (Figures 2–4). In order to avoid the computationally expensive task of computing cardinality-minimal unsatisfiable subsets, we approximated the number of iterative local backbones by computing subset-minimal unsatisfiable subsets, using the MUSer2 algorithm [Belov and Marques-Silva 2012]. We considered instances originating from planning [Hoos and Stützle 2000; Kautz and Selman 1996], circuit fault analysis [Prelotani 1996], and bounded model checking [Strichman 2000]. For each of the instances, we give the percentage of backbones that are of order k (dashed lines) and the percentage of backbones that are of iterative order k (solid lines), as well as the total number of backbones and the total number of clauses. For these instances, most backbones are (iterative) local backbones of very low order. For example, already more than 75 percent of the backbones in all the considered *bmc-ibm* instances are of iterative order 2.

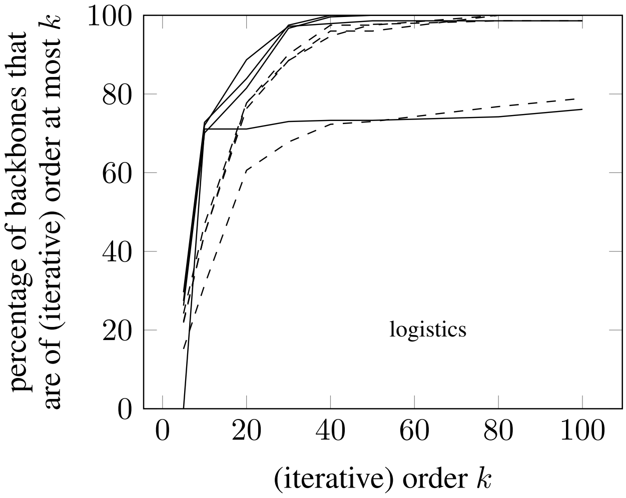


Fig. 2: Percentage of backbones that are of order at most k (dashed) and of iterative order at most k (solid), for SAT instances from planning (*logistics.[a–d]*, 828–4713 variables, 6718–21991 clauses, 437–838 backbones).

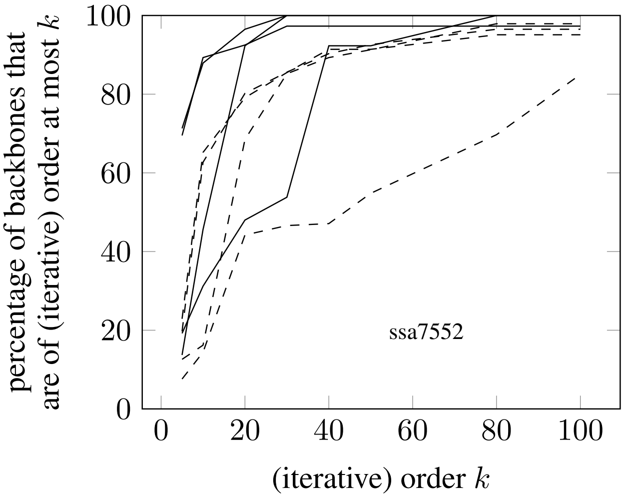


Fig. 3: Percentage of backbones that are of order at most k (dashed) and of iterative order at most k (solid), for SAT instances from circuit fault analysis (*ssa7552-[038,158–160]*, 1363–1501 variables, 3032–3575 clauses, 405–838 backbones).

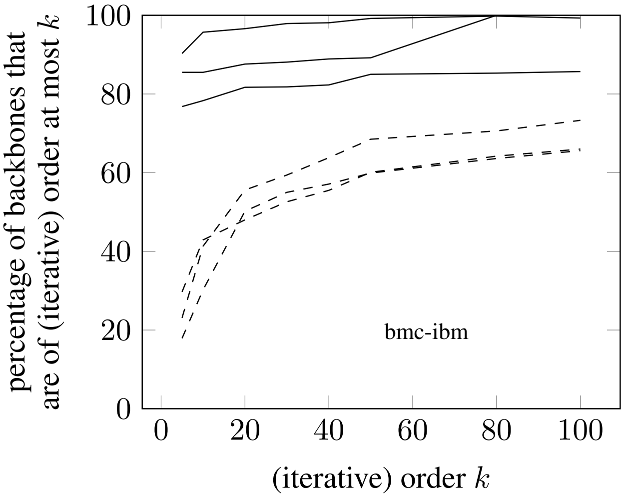


Fig. 4: Percentage of backbones that are of order at most k (dashed) and of iterative order at most k (solid), for SAT instances from bounded model checking (*bmc-ibm-[2,5,7]*, 2810–9396 variables, 11683–41207 clauses, 405–557 backbones).