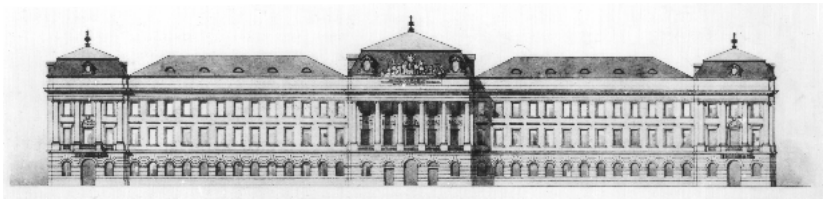
**INFSYS**
**RESEARCH**
**REPORT**



INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG WISSENSBASIERTE SYSTEME

# SMALL UNSATISFIABLE SUBSETS IN CONSTRAINT SATISFACTION

Ronald de Haan Iyad Kanj Stefan Szeider

INFSYS RESEARCH REPORT 1843-14-05

DECEMBER 2014

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT
INFSYS RESEARCH REPORT 1843-14-05, DECEMBER 2014

# SMALL UNSATISFIABLE SUBSETS IN CONSTRAINT SATISFACTION

Ronald de Haan[1*] Iyad Kanj[2] Stefan Szeider[1*]

**Abstract.** The problem of finding small unsatisfiable subsets of a set of constraints is important for various applications in computer science and artificial intelligence. We study the problem of identifying whether a given instance to the constraint satisfaction problem (CSP) has an unsatisfiable subset of size at most $k$ from a parameterized complexity point of view. We show that the problem of finding small unsatisfiable subsets of a CSP instance is harder than the corresponding problem for CNF formulas. Moreover, we show that the problem is not fixed-parameter tractable when restricting the problem to any maximal tractable Boolean constraint language (for which the problem is nontrivial). We show that the problem is hard even when the maximum number of occurrences of any variable is bounded by a constant, a restriction which leads to fixed-parameter tractability for the case of CNF formulas. Finally, we relate the problem of finding small unsatisfiable subsets to the problem of identifying variable assignments that are enforced already by a small number of constraints (backbones), or that are ruled out already by a small number of constraints (anti-backbones).

[1] Institute of Information Systems, Vienna University of Technology, Vienna, Austria
[2] School of Computing, DePaul University, Chicago, IL
[*] Supported by the European Research Council (ERC), project 239962 (COMPLEX REASON), and the Austrian Science Fund (FWF), project P26200 (Parameterized Compilation).

**Publication Information:** This is the authors' self-archived copy of a paper that appeared in the *Proceedings of the 26th International Conference on Tools with Artificial Intelligence (ICTAI 2014)*.

# 1 Introduction

Finding small or unsatisfiable cores (subsets) of an unsatisable set of constraints is useful for a variety of purposes, such as verifying the correctness of a solver or computational tasks arising in various applications (formal verification, repairing inconsistencies in knowledge bases, maximal satisfiability solving, etc). There has been much research on the topic of developing fast algorithms to find minimal (subset-minimal) or minimum (cardinality-minimal) unsatisfiable cores for propositional formulas (cf. [2, 1, 3]), possibly over an underlying theory (SMT) [4], as well as for instances of the Constraint Satisfaction Problem (CSP) [5]. We focus on the CSP, and we investigate the problem of finding small unsatisfiable subsets of constraints from a parameterized complexity point of view. This extends the research of Fellows, Szeider and Wrightson [6] and De Haan, Kanj and Szeider [7] on the parameterized complexity of identifying small unsatisfiable subsets in propositional formulas.

The problem consists of deciding, given a finite set of constraints and a positive integer $k$, whether there exists a subset of $k$ many constraints that is unsatisfiable. It is straightforward to devise an xp-algorithm to solve the problem (i.e., an algorithm that runs in time $O(n^{f(k)})$, where $n$ is the input size). This can, for instance, be done by iterating over all possible subsets of $k$ many constraints, and for each subset checking satisfiability in a brute-force manner. However, to develop practical algorithms, it would be much more desirable to identify in what cases the problem becomes *fixed-parameter tractable* (i.e., solvable in time $f(k)n^c$, where $f$ is some computable function and $c$ is some fixed constant). This important distinction between xp-solvability and fixed-parameter tractability is the focus of parameterized complexity theory [10, 11, 12, 13], a field of research that is becoming increasingly popular in the domain of artificial intelligence and constraint satisfaction (cf. [14, 15, 16]).

Unsurprisingly, the problem of finding an unsatisfiable subset in its most general setting is not fixed-parameter tractable. A special case of the problem is to identify whether a 3CNF formula has an unsatisfiable subset of size $k$, and this problem is already W[1]-hard [6], that is, it is not fixed-parameter tractable, under complexity-theoretic assumptions. For this reason, we direct our attention to several restrictions of the problem for which it can be decided in polynomial time whether a given set of constraints is satisfiable or not. In particular, we focus on several classes of Boolean constraint languages that have been identified in the seminal work of Schaefer [17] as the maximal constraint languages for which the satisfiability problem is polynomial-time tractable.

For an overview of the parameterized complexity results obtained in this paper, see Table 1. Interestingly, the problem of identifying small unsatisfiable subsets of CSP instances is fixed-parameter *in*tractable (W[1]-hard or W[2]-hard) for all nontrivial restrictions that we consider. 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. For the unrestricted case, the problem of identifying small unsatisfiable subsets is harder when dealing with CSP instances (W[2]-hard; Proposition 1) than when dealing with propositional formulas (W[1]-complete [6]). Similarly, for the restriction to Krom formulas and correspondingly bijunctive constraint languages, the problem increases in complexity when moving from propositional formulas (poly-time [9]) to CSP instances (W[1]-hard; Theorem 5). The problem is W[1]-hard

| Boolean language $\mathcal{C}$ | SMALL-UNSAT-SUBSET[$\mathcal{C}$] | | SMALL-CSP-UNSAT-SUBSET[$\mathcal{C}$] | |
|---|---|---|---|---|
| unrestricted | W[1]-hard | [6] | W[2]-hard | (Prop 1) |
| 0-valid, 1-valid | trivial | (Obs 2) | trivial | (Obs 2) |
| Horn, anti-Horn | W[1]-hard | [7] | W[1]-hard | (Prop 3, Cor 4) |
| bijunctive (Krom) | poly-time | [9] | W[1]-hard | (Thm 5) |
| affine | W[1]-hard | (Thm 7) | W[1]-hard | (Cor 6) |
| bounded degree | FPT | [7] | W[1]-hard | (Prop 8) |

Table 1: Map of parameterized complexity results for the problems SMALL-UNSAT-SUBSET and SMALL-CSP-UNSAT-SUBSET. For each Boolean language $\mathcal{C}$, the table contains results for the restriction of the problems to the corresponding class of propositional formulas and Boolean constraint languages, respectively. (We slightly abuse notation: technically, the class of CSP instances with bounded degree is not generated by any constraint language.)

both for the case of affine propositional formulas and for the case of affine constraint languages (Corollary 6, Theorem 7). Upper bounding the degree (i.e., the maximum number of occurrences of any variable) by a constant leads to fixed-parameter tractability in the case of propositional formulas [6, 7]. However, in the case of constraint satisfaction, the problem of finding small unsatisfiable subsets remains W[1]-hard even when restricted to Boolean CSP instances of bounded degree (Proposition 8). Finally, the problem of finding small unsatisfiable subsets is trivially tractable when restricted to 0-valid or 1-valid constraint languages. However, for the related problem of identifying variable assignments that are implied by a small subset of constraints (see Section 4), the restriction to 0-valid or 1-valid constraint languages yields W[2]-hardness (Proposition 12). We would like to point out that the differences in complexity between the cases of propositional formulas and Boolean CSP instances are not related to the domain size, since in both cases the domain is Boolean.

# 2 Preliminaries

## 2.1 Parameterized Complexity

We introduce the relevant concepts of parameterized complexity theory. For more details, we refer to textbooks on the topic [10, 13, 11, 12]. 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 $f(k)|I|^c$, where $f$ is a computable function of $k$, and $c$ is a constant (algorithms running within such time bounds are called *fpt-algorithms*). 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. 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 $f(k)|I|^c$ for a computable function $f$ and a constant $c$.

Central to the completeness theory is the hierarchy FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP. Each intractability class W[t] contains all parameterized problems that can be reduced to a certain parameterized satisfiability problem under fpt-reductions. 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. Fixed-parameter tractability of any problem hard for any of these intractability classes is very unlikely as it would violate commonly-believed assumptions in complexity theory, such as the Exponential Time Hypothesis [11, 18] (i.e., the existence of a $2^{o(n)}$ algorithm for $n$-variable 3SAT).

We use the following problems to prove some fixed-parameter intractability results. MULTI-COLORED-CLIQUE is a W[1]-complete problem [19]. The instances are tuples $(V,E,k)$, where $V$ is a finite set of vertices partitioned into $k$ subsets $V_1,\ldots,V_k$, $(V,E)$ 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 [20]. 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 $T \in \mathcal{T}$.

## 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 $\varphi$ 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. A clause is a *Horn clause* if it contains at most one positive literal. CNF formulas containing only Horn clauses are called *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 $\varphi$ is the number of clauses of $\varphi$ in which it occurs (positively or negatively). The degree of $\varphi$ is the maximum degree of any variable that occurs in $\varphi$. 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 $\varphi$ 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 $\tau$ satisfies $\varphi$); otherwise, $\varphi$ 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. An *affine* formula is a finite set of affine clauses, and is interpreted as the conjunction of these clauses. An affine formula $\varphi$ is a $k$-affine formula if the size of each of its affine clauses is at most $k$.

An affine formula $\varphi$ is *satisfiable* if there exists a truth assignment $\tau : \mathrm{Var}(\varphi) \to \{0,1\}$ such that every clause $c \in \varphi$ contains an odd number of literals $l$ such that $\tau(l) = 1$. (we say that such an assigment $\tau$ satisfies $\varphi$); otherwise, $\varphi$ is *unsatisfiable*.

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

We denote the parameterized problem of identifying whether a given CNF or affine formula $\varphi$ has an unsatisfiable subset of size at most $k$, parameterized by $k$, by SMALL-UNSAT-SUBSET. Moreover, for a given set $\mathcal{C}$ of formulas, we denote the restriction of this problem to formulas in $\mathcal{C}$ by SMALL-UNSAT-SUBSET$[\mathcal{C}]$.

## 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 $S = (v_1, \ldots, v_n)$ is a sequence of variables from $\mathcal{V}$ and $R \in \Gamma$ is a relation in the constraint language $\Gamma$ (called the *constraint relation*). The set $\mathrm{Var}(C) = \{v_1, \ldots, v_n\}$ 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 $C = ((v_1, \ldots, v_n), R)$ if $\mathrm{Var}(C) \subseteq V$ and $(\alpha(v_1), \ldots, \alpha(v_n)) \in R$. For a set $\mathcal{I}$ of constraints, we write $\mathrm{Var}(\mathcal{I}) = \bigcup_{C \in \mathcal{I}} \mathrm{Var}(C)$, and we write $\mathrm{Rel}(\mathcal{I}) = \{\, R : (S, R) \in C, C \in \mathcal{I} \,\}$. 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 $\mathcal{I}$.

An assignment $\alpha : \mathrm{Var}(\mathcal{I}) \to D$ 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 $\mathrm{CSP}(\Gamma)$ we denote the CSP restricted to instances $\mathcal{I}$ with $\mathrm{Rel}(\mathcal{I}) \subseteq \Gamma$. A constraint language is *tractable* if for every finite subset $\Gamma' \subseteq \Gamma$, the problem $\mathrm{CSP}(\Gamma)$ can be solved in polynomial time.

We call a constraint language $\Gamma$ *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 $(0, \ldots, 0) \in R$;

- *1-valid* if $(1, \ldots, 1) \in R$;
- *Horn* if $R$ is equivalent to a CNF formula that is 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 $\Gamma$ is 0-valid, 1-valid, Horn, anti-Horn, bijunctive, affine or 2-affine, respectively, if all relations $R \in \Gamma$ have this property.

In his seminal paper [17], Schaefer showed that for all constraint languages $\Gamma$ over the Boolean domain $\{0, 1\}$, the CSP restricted to $\Gamma$ is either NP-complete or solvable in polynomial time. In fact, he showed that a Boolean constraint language $\Gamma$ 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*.

Let $\alpha : X \to \mathcal{D}$ be an assignment. For an $n$-ary constraint $C = (S, R)$ with $S = (x_1, \ldots, x_n)$ we denote by $C|_\alpha$ the constraint $(S', R')$ obtained from $C$ as follows. $R'$ is obtained from $R$ by (i) deleting all tuples $(d_1, \ldots, d_n)$ 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 $\Gamma$ is *closed under partial assignment* if for any constraint $C$ over $\Gamma$ and any assignment $\alpha : X \to \mathcal{D}$ it holds that $C|_\alpha$ is also a constraint over $\Gamma$.

# 3 Small Unsatisfiable Subsets for the CSP

Consider the decision problem of finding an unsatisfiable subset of a given CSP instance. It is well-known that this problem is co-NP-complete in general (since this involves deciding whether the CSP instance is unsatisfiable), but polynomial-time solvable when restricted to one of the Schaefer languages. In this section, we consider the following parameterization of this problem, for various constraint languages.

SMALL-CSP-UNSAT-SUBSET
*Instance:* a CSP instance $\mathcal{I}$, and a positive integer $k$.
*Parameter:* $k$.
*Question:* Is there a subset $\mathcal{I}'$ of $\mathcal{I}$ with $k$ many constraints that is unsatisfiable?

## 3.1 Complexity Results for Boolean Domains

**Proposition 1.** SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances 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 $\mathcal{T} = \{T_1, \ldots, T_m\}$ is a family of subsets of the universe $U = \{u_1, \ldots, u_n\}$. The idea behind this reduction is the following. We introduce a variable for each subset $T_j \in \mathcal{T}$. Moreover, we have one constraint $C_0$ that ensures that exactly one of these variables is assigned the value 1. Then, for each element $u$ of the universe $U$ we add a constraint that ensures that all variables corresponding to subsets $T_j \in \mathcal{T}$ containing this element $u$ are assigned the value 0. Any unsatisfiable subset of constraints must then include both the constraint $C_0$, and some other constraints that correspond to a hitting set.

We construct a CSP instance $\mathcal{I}$ over the domain $D = \{0, 1\}$, as follows. We let the set $\mathrm{Var}(\mathcal{I}) = \{ v_1, \ldots, v_m \}$ of variables correspond to the universe $U$. Firstly, we add the constraint $C_0 = (S_0, R_0)$ to $\mathcal{C}$, with scope $S_0 = (v_1, \ldots, v_m)$, and where $R_0 = \{(1, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)\}$. Then, for each $u_i \in U$, we add a constraint $C_i = (S_i, R_i)$ to $\mathcal{C}$. We consider $W(u_i) = \{ T_j : 1 \leq j \leq m, u_i \in T_j \}$ and we write $W(u_i) = \{u_{i_1}, \ldots, u_{i_\ell}\}$. We then let $S_i = (v_{i_1}, \ldots, v_{i_\ell})$, and we let $R_i = \{(0, \ldots, 0)\}$. 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.

($\Rightarrow$) Suppose that there exist $1 \leq i_1 < \ldots < i_k \leq n$ such that $U' = \{u_{i_1}, \ldots, u_{i_k}\}$ is a hitting set of $\mathcal{T}$. Now consider the subset $\mathcal{I}' = \{C_0\} \cup \{ C_{i_j} : 1 \leq j \leq k \}$. Clearly, $\mathcal{I}'$ has $k'$ many constraints, and it is straightforward to verify that $\mathcal{I}'$ is unsatisfiable.

($\Leftarrow$) Assume that there exists some subset $\mathcal{I}'$ of $\mathcal{I}$ that has $k'$ many constraints and that is unsatisfiable. Then $C_0 \in \mathcal{I}'$, because otherwise setting all variables to 0 would satisfy $\mathcal{I}'$. This can only be the case if $\mathcal{I}'$ includes $k$ many constraints $C_i$ that together force all variables $v \in V$ to get the value 0. This would correspond to a hitting set of $\mathcal{T}$ of size $k$. □

**Observation 2.** 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-UNSAT-SUBSET *can be solved (trivially) in polynomial time when restricted to CNF formulas that are* 0*-valid or* 1*-valid.*

*Proof.* Any Boolean CSP instance that is 0-valid or 1-valid is clearly satisfiable, so the problem SMALL-CSP-UNSAT-SUBSET restricted to 0-valid and 1-valid CSP instances is trivial. Similarly, CNF formulas that are 0-valid or 1-valid are clearly satisfiable, and thus the same argument holds for SMALL-UNSAT-SUBSET. □

**Proposition 3.** SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances that are Horn is* W[1]*-hard.*

*Proof.* We know that the problem of identifying whether a given propositional formula that is Horn and in 3CNF has an unsatisfiable subset of size at most $k$ is W[1]-hard [7]. Since any propositional formula that is Horn and in 3CNF can be equivalently expressed as a Boolean CSP instance that is Horn (where each clause is expressed by a single constraint), the W[1]-hardness result follows. □

**Corollary 4.** SMALL-CSP-UNSAT-SUBSET *restricted to Boolean CSP instances that are anti-Horn is* W[1]*-hard.*

*Proof.* We know that consistency of CSP instances is invariant under swapping of domain values. Then, by swapping the two values 0 and 1, we can transform any Boolean Horn CSP instance into an equivalent anti-Horn CSP instance. □

For the case of propositional formulas, the problem of identifying small unsatisfiable subsets becomes fixed-parameter tractable when restricted to formulas where the degree (i.e., the maximum number of times that any variable appears in the formula) is bounded by a constant [6, 7]. We consider a similar restriction for CSP instances. Let $\mathcal{I}$ be a CSP instance. For any $v \in \text{Var}(\mathcal{I})$, we define the *degree* of $v$ to be the number of constraints where $v$ appears in the scope. Moreover, we let the *degree* of $\mathcal{I}$ be the maximum degree of any variable $v \in \text{Var}(\mathcal{I})$. Unlike the case of propositional formulas, it turns out that the problem restricted to instances whose degree is bounded by a constant is W[1]-hard.

**Theorem 5.** SMALL-CSP-UNSAT-SUBSET *restricted to bijunctive Boolean CSP instances 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 $V_1, \ldots, V_k$. We construct a bijunctive Boolean CSP instance $\mathcal{I}$, and an integer $k'$. We let $\text{Var}(\mathcal{I})$ 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 $e = (v_i, v_j) \in (V_i \times V_j) \cap E$, for $i < j$, we introduce the constraint $C_e = (S_e, R_e)$, where $S_e = (x_{v_i,j}, x_{v_i,j+1}, x_{v_i,j}, x_{v_j,i+1})$ and where $R_e$ is equivalent to the following Krom formula:

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

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

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

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

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

Finally, for each $v_1 \in V_1$ and each $v_k \in V_k$, we introduce the constraint $C_{v_k,v_1} = (S_{v_k,v_1}, R_{v_k,v_1})$, where $S_{v_k,v_1} = (z_0, x_{v_1,1}, x_{v_k,k+1})$ and where $R_{v_k,v_1}$ is equivalent to the following Krom formula:

$$\begin{aligned} R_{v_k,v_1} \equiv\ & (z_0 \leftrightarrow x_{v_1,1}) \wedge (x_{v_k,k+1} \leftrightarrow \overline{z_0}) \\ \equiv\ & (z_0 \rightarrow x_{v_1,1}) \wedge (x_{v_1,1} \rightarrow z_0) \wedge \\ \equiv\ & (x_{v_k,k+1} \rightarrow \overline{z_0}) \wedge (\overline{z_0} \rightarrow x_{v_k,k+1}). \end{aligned}$$

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 $\{v_1, \ldots, v_k\} \subseteq V$ of vertices such that for each $1 \leq i \leq k$, $v_i \in V_i$, and for each $1 \leq i < i' \leq k$, $(v_i, v_{i'}) \in E$. Consider the subset $\mathcal{I}' \subseteq \mathcal{I}$ consisting of the following constraints:

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

It is easy to verify that $\mathcal{I}'$ consists of $k'$ many constraints. Moreover, it is straightforward to verify that any solution $\alpha$ 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 $C_{v_k, v_1}$, for some $v_k \in V_k$ and some $v_1 \in V_1$. 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 $(x_{v_1,1} \leftrightarrow x_{v_k,k+1})$; otherwise $\mathcal{I}'$ would be satisfiable. Then we also know that $\mathcal{I}'$ must include, for each $1 \leq i < k$, a constraint $C_{v_i, v_{i+1}}$ for some $v_i \in V_i$ and some $v_{i+1} \in V_{i+1}$; otherwise, the equivalence $(x_{v_1,1} \leftrightarrow x_{v_k,k+1})$ would not be enforced. Finally, $\mathcal{I}'$ must enforce the equivalences $(x_{v_i,1} \leftrightarrow x_{v_i,k+1})$, 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 $C_{v_i}$ for each $1 \leq i \leq k$, and the constraints $C_{e_{ij}}$ for each $e_{ij} = (v_i, v_j)$, 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 $\{v_1, \ldots, v_k\}$ is a multi-colored $k$-clique of $G$. Therefore, $(G, k) \in$ MULTI-COLORED-CLIQUE. □

**Corollary 6.** 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 Theorem 5. 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. □

**Theorem 7.** 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 Theorem 5. Let $(G, k)$ be an instance of MULTI-COLORED-CLIQUE, where $G = (V, E)$ and where $V$ is partitioned into $V_1, \ldots, V_k$. 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 $e = (v_i, v_j) \in (V_i \times V_j) \cap E$, for $1 \leq i < j \leq k$, we introduce the constraint $C_e = (S_e, R_e)$, where $S_e = (x_{v_i,j}, x_{v_i,j+1}, x_{v_i,j}, x_{v_j,i+1})$ and where $R_e$ is equivalent to the following affine clause:

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

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

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

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

$$R_{v_i,v_{i+1}} \equiv \neg(x_{v_i,k+1} \oplus x_{v_{i+1},1}) \equiv (\overline{x_{v_i,k+1}} \oplus x_{v_{i+1},1}).$$

Finally, for each $v_1 \in V_1$ and each $v_k \in V_k$, we introduce the constraint $C_{v_k,v_1} = (S_{v_k,v_1}, R_{v_k,v_1})$, where $S_{v_k,v_1} = (z_0, x_{v_1,1}, x_{v_k,k+1})$ and where $R_{v_k,v_1}$ is equivalent to the following affine clause:

$$\begin{aligned} R_{v_k,v_1} \equiv & \quad \neg(z_0 \oplus x_{v_1,1} \oplus x_{v_k,k+1} \oplus z_1) \\ \equiv & \quad (\overline{z_0} \oplus x_{v_1,1} \oplus x_{v_k,k+1} \oplus z_1). \end{aligned}$$

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 $\{v_1, \ldots, v_k\} \subseteq V$ of vertices such that for each $1 \leq i \leq k$, $v_i \in V_i$, and for each $1 \leq i < i' \leq k$, $(v_i, v_{i'}) \in E$. Consider the subset $\mathcal{I}' \subseteq \mathcal{I}$ consisting of the following constraints:

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

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

$$\begin{aligned}\sigma = &\ (l_1, \ldots, l_m) \\ = &\ (z_0, x_{v_1,1}, \ldots, x_{v_1,k+1}, x_{v_2,1}, \ldots, x_{v_k,k+1}, z_1).\end{aligned}$$

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 $\alpha$ 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 $\alpha$ 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 Theorem 5, 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. □

**Proposition 8.** *Let $d \geq 2$ be a constant.* SMALL-CSP-UNSAT-SUBSET *is* W[1]*-hard, even when restricted to Boolean CSP instances with degree $\leq d$.*

*Proof.* We know that the problem of finding an unsatisfiable subset of a propositional formula of size $\leq k$ 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 $\varphi = \{c_1, \ldots, c_m\}$ be a propositional formula in 3CNF, and let $k$ be a positive integer. Without loss of generality, we may assume that $\varphi$ contains no unsatisfiable subsets of size $< k$, that any unsatisfiable subset of $\varphi$ of size $k$ contains exactly $\ell$ variables, where $\ell$ can be computed in polynomial time [7, Lemmas 2 and 3 and Theorem 2].

We now construct an instance $(\mathcal{I}, k')$ over the domain $D$ of SMALL-CSP-UNSAT-SUBSET as follows. We let 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 satisfying $c$. Next, for each variable $x \in$ Var($\varphi$), we add a constraint $C_x = (S_x, R_x)$ to $\mathcal{I}$. We let $S_x = (v_{x,c_1}, \ldots, v_{x,c_m})$, and we let $R_x = \{(0, \ldots, 0), (1, \ldots, 1)\}$. Finally, we define $k' = k + \ell$. It is straightforward to verify that $\mathcal{I}$ has degree 2.

We now claim that $\varphi$ has an unsatisfiable subset of size $k$ if and only if $\mathcal{I}$ has an unsatisfiable subset of size $k'$.

($\Rightarrow$) Assume that $\varphi$ has an unsatisfiable subset of size $k$. Let $\varphi' = \{c'_1, \ldots, c'_k\}$ be such a subset. We know that exactly $\ell$ variables appear in $\varphi'$. 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 $\varphi$ of size $< k$, which is a contradiction. Thus, $k_1 \geq k$. Next, suppose that $k_2 < \ell$. Then, we know that $\varphi'$ is an unsatisfiable subset of $\varphi$. However, by minimality of $\mathcal{I}'$, we then know that $\varphi'$ 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 $k_2 = \ell$. It is now straightforward to verify that $\varphi'$ is an unsatisfiable subset of $\varphi$ of size $k$. □

## 3.2 Complexity Results for Non-Boolean Domains

When we lift our restriction to Boolean domains, the problem of identifying a small unsatisfiable subset of a CSP instance is W[2]-hard already when restricted to unary constraints. Note that in this case, the domain $D$ is given as part of the input.

**Proposition 9.** 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 $\mathcal{T} = \{T_1, \ldots, T_m\}$ is a family of subsets of the universe $U = \{u_1, \ldots, u_n\}$. The idea behind this reduction is the following. We introduce a single variable, and we introduce one domain element for each subset $T_j \in \mathcal{T}$. Moreover, we introduce a single constraint for each element $u$ in the universe $U$. Intuitively, the constraint for element $u$ rules out that the variable is assigned values 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 $\text{Var}(\mathcal{I}) = \{v\}$ consist of a single variable, and we let its domain $D = \{d_1, \ldots, d_m\}$ 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 at most $k$ of these constraints then corresponds exactly to finding a hitting set of size at most $k$. 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 $1 \leq i_1 < \cdots < i_\ell \leq n$ such that $C_{i_j} \in \mathcal{I}'$ for all $1 \leq j \leq \ell$. We claim that $U' = \{u_{i_1}, \ldots, u_{i_\ell}\}$ 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 $\alpha$ 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 $1 \leq i_1 < \cdots < i_\ell \leq n$ such that $U' = \{u_{i_1}, \ldots, u_{i_\ell}\}$ 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 $\alpha$ 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 $\mathcal{T}$. □

# 4 Local Backbones and Anti-Backbones

A concept related to unsatisfiable subsets are *backbones*. A backbone (with truth value $d$) of a propositional formula $\varphi$ is a variable $v \in \mathrm{Var}(\varphi)$ that is assigned value $d$ in all solutions of $\varphi$. The term originates in computational physics [21], and the notion of backbones has been studied for propositional formulas in various contexts. Backbones have also been considered in other contexts (e.g., knowledge compilation [22]) and for other combinatorial problems [23].

In a similar way, we can define backbones of CSP instances. A backbone (with value $d$) of a CSP instance $\mathcal{I}$ is a variable $v \in \mathrm{Var}(\mathcal{I})$ that is assigned value $d$ in all solutions of $\mathcal{I}$. For constraint satisfaction, one could also consider the (dual) notion of *anti-backbones*. An anti-backbone (with value $d$) of a CSP instance $\mathcal{I}$ is a variable $v \in \mathrm{Var}(\mathcal{I})$ that is assigned value $d$ in no solution of $\mathcal{I}$.

If a backbone and its value are known, or if an anti-backbone and its value are known, then we can simplify the instance without removing solutions, or the number of solutions. Therefore, it is desirable to have an efficient algorithm for detecting backbones and anti-backbones. In general, however, it is straightforward to verify that the problem of identifying backbones or anti-backbones is coNP-complete.

A variable can be a backbone or an anti-backbone because of *local properties* of the instance (such (anti-)backbones we call *local backbones* and *local anti-backbones*). As an extreme example consider a CSP instance that contains a unary constraint with only one tuple $(d)$ (a *unit constraint*). In this case we know that the variable $v$ appearing in this constraint clause is a backbone of the CSP instance (with value $d$). Additionally, the variable $v$ is an anti-backbone of the CSP instance (with any value $d' \neq d$). More generally, we define the *order* of a backbone $x$ of a CSP instance $\mathcal{I}$ to be the cardinality of a smallest subset $\mathcal{I}'$ of constraints such that $x$ is a backbone of $\mathcal{I}'$, and we refer to backbones of order $\leq k$ as *$k$-backbones*. We define the notion of order for anti-backbones and the notion of *$k$-anti-backbones* similarly. Thus, unit constraints give rise to 1-backbones and 1-anti-backbones for Boolean CSP instances. Now consider the following parameterized decision problem, concerned with the identification of local (anti-)backbones.

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

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

We investigate the parameterized complexity of these problems, restricted to the various Schaefer languages that we also considered in the previous section. We firstly show that these problems are at least as hard as the problem SMALL-CSP-UNSAT-SUBSET, for all constraint languages.

**Proposition 10.** *The problem* SMALL-CSP-UNSAT-SUBSET *is fpt-reducible to the problems* LOCAL-CSP-BACKBONE *and* LOCAL-CSP-ANTI-BACKBONE*.*

*Proof (sketch).* We first give an fpt-reduction to LOCAL-CSP-BACKBONE. The main idea behind this reduction is that 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 $v$ to $\mathcal{I}$ (for instance, by adding an additional constraint $C$ with $\text{Var}(C) = \{v\}$). We may assume without loss of generality that $\{0, 1\} \subseteq D$, and that the variable $v$ is unconstrained and can get either value 0 or 1 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 $\alpha$ to satisfy $\alpha(v) = 0$.

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 $\alpha$ to satisfy $\alpha(v) \neq 0$. □

Most Schaefer languages are closed under partial assignment, i.e., the Horn, anti-Horn, bijunctive and affine languages. For these languages, the complexity of these problems is closely related to the complexity of SMALL-CSP-UNSAT-SUBSET.

**Proposition 11.** *When restricted to Boolean constraint languages that are closed under partial assignment, the problems* LOCAL-CSP-BACKBONE *and* LOCAL-CSP-ANTI-BACKBONE *are both fpt-reducible to* SMALL-CSP-UNSAT-SUBSET*.*

*Proof (sketch).* We firstly give an fpt-reduction for the case of LOCAL-CSP-BACKBONE. Let $(\mathcal{I}, v, d, k)$ be an instance of LOCAL-CSP-BACKBONE. We construct an equivalent instance of SMALL-CSP-UNSAT-SUBSET as follows. We define the assignment $\alpha : \{v\} \to \mathcal{D}$ by letting $\alpha(v) = d'$, where $d'$ is the unique value in $\{0, 1\} \backslash \{d\}$. We then let $\mathcal{I}' = \mathcal{I}|_\alpha$. It is straightforward to verify that $(\mathcal{I}', k)$ is a yes-instance of SMALL-CSP-UNSAT-SUBSET if and only if $(\mathcal{I}, v, d, k)$ is a yes-instance of LOCAL-CSP-BACKBONE.

A similar fpt-reduction works for the case of LOCAL-CSP-ANTI-BACKBONE, with the difference that $\alpha(v) = d$. □

The Schaefer languages that are not conservative are the 0-valid and 1-valid constraint languages. It turns out that the problems LOCAL-CSP-BACKBONE and LOCAL-CSP-ANTI-BACKBONE are W[2]-hard when restricted to 0-valid or 1-valid Boolen CSP instances. This is in contrast with the 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.

**Proposition 12.** LOCAL-CSP-BACKBONE *is* W[2]*-hard, when restricted to Boolean CSP instances that are* 0*-valid.*

*Proof (sketch).* The proof of Proposition 1 can be straightforwardly modified to show this result. The idea of this modification is to introduce an additional variable that is forced to be assigned the value 0 if and only if the instance would contain an unsatisfiable subset otherwise.

We modify the definition of $C_0$ as follows. We introduce an additional variable $v_0$, and we let $S_0 = (v_0, v_1, \ldots, v_n)$. We let $R_0 = \{(1,1,0,\ldots,0),(1,0,1,0,\ldots,0),\ldots,(1,0,\ldots,0,1),(0,\ldots,0)\}$. It is straightforward to verify that the constructed instance $\mathcal{I}'$ now is 0-valid. Moreover, it can straightforwardly be verified that there is a subset $\mathcal{I}' \subseteq \mathcal{I}$ of $k'$ many constraints that enforces that $v_0$ is assigned value 0 if and only if $(U, \mathcal{T}, k) \in$ HITTING-SET. □

**Corollary 13.** LOCAL-NECESSARY-ASSIGNMENT *and* LOCAL-IMPOSSIBLE-ASSIGNMENT *are* W[2]*-hard, when restricted to Boolean CSP instances that are* 1*-valid.*

# 5 Conclusion

We studied the problem of identifying whether a given instance of the constraint satisfaction problem has an unsatisfiable subset of size at most $k$ from a parameterized complexity point of view. We showed that the problem is W[1]-hard or even harder when restricting the problem to any maximal tractable Boolean constraint language (for which the problem is nontrivial), and that the problem is hard even when the degree of instances is bounded by a constant. For non-Boolean domains, we showed that the problem is already W[2]-hard when restricted to unary constraints. We also related the problem of finding a small unsatisfiable subset to the problem of identifying local backbones and local anti-backbones, and showed that these latter problems are of the same parameterized complexity (or even harder).

Future research includes investigating the relation between local backbones and anti-backbones and local consistency notions, such as arc consistency and hyper-arc consistency.

# References

[1] I. Lynce and J. P. M. Silva, "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)*, 2004.

[2] A. Belov, I. Lynce, and J. Marques-Silva, "Towards efficient MUS extraction," *AI Commun.*, vol. 25, no. 2, pp. 97–116, 2012.

[3] J. Marques-Silva, "Computing minimally unsatisfiable subformulas: State of the art and future directions," *Journal of Multiple-Valued Logic and Soft Computing*, pp. 163–183, 2012.

[4] A. Cimatti, A. Griggio, and R. Sebastiani, "Computing small unsatisfiable cores in satisfiability modulo theories." *J. Artif. Intell. Res.*, vol. 40, pp. 701–728, 2011.

[5] M. H. Liffiton and K. A. Sakallah, "Algorithms for computing minimal unsatisfiable subsets of constraints," *Journal of Automated Reasoning*, vol. 40, no. 1, pp. 1–33, 2008.

[6] M. R. Fellows, S. Szeider, and G. Wrightson, "On finding short resolution refutations and small unsatisfiable subsets," *Theoretical Computer Science*, vol. 351, no. 3, pp. 351–359, 2006.

[7] R. de Haan, I. Kanj, and S. Szeider, "Local backbones," arXiv.org, Tech. Rep. 1304.5479, 2013, (updated version of [8]).

[8] R. de Haan, I. Kanj, and S. Szeider, "Local backbones," in *Proceedings of SAT 2013, 16th International Conference on Theory and Applications of Satisfiability Testing, July 8-12, 2013, Helsinki, Finland*, ser. Lecture Notes in Computer Science, M. Järvisalo and A. V. Gelder, Eds., vol. 7962. Springer Verlag, 2013, pp. 377–393.

[9] J. Buresh-Oppenheim and D. G. Mitchell, "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*, ser. Lecture Notes in Computer Science, A. Biere and C. P. Gomes, Eds., vol. 4121, 2006, pp. 42–47.

[10] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, ser. Monographs in Computer Science. New York: Springer Verlag, 1999.

[11] J. Flum and M. Grohe, *Parameterized Complexity Theory*, ser. Texts in Theoretical Computer Science. An EATCS Series. Berlin: Springer Verlag, 2006, vol. XIV.

[12] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, ser. Oxford Lecture Series in Mathematics and its Applications. Oxford: Oxford University Press, 2006.

[13] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, ser. Texts in Computer Science. Springer Verlag, 2013.

[14] G. Gottlob and S. Szeider, "Fixed-parameter algorithms for artificial intelligence, constraint satisfaction, and database problems," *The Computer Journal*, vol. 51, no. 3, pp. 303–325, 2006, survey paper.

[15] M. Samer and S. Szeider, "Constraint satisfaction with bounded treewidth revisited," *J. of Computer and System Sciences*, vol. 76, no. 2, pp. 103–114, 2010.

[16] S. Gaspers and S. Szeider, "The parameterized complexity of local consistency," in *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP 2011)*, ser. Lecture Notes in Computer Science, J. H.-M. Lee, Ed., vol. 6876. Springer Verlag, 2011, pp. 302–316.

[17] T. J. Schaefer, "The complexity of satisfiability problems," in *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978)*. ACM, 1978, pp. 216–226.

[18] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *J. of Computer and System Sciences*, vol. 63, no. 4, pp. 512–530, 2001.

[19] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette, "On the parameterized complexity of multiple-interval graph problems," *Theoretical Computer Science*, vol. 410, no. 1, pp. 53–61, 2009.

[20] R. G. Downey and M. R. Fellows, "Fixed-parameter tractability and completeness. II. On completeness for $W[1]$," *Theoretical Computer Science*, vol. 141, no. 1-2, pp. 109–131, 1995.

[21] J. Schneider, C. Froschhammer, I. Morgenstern, T. Husslein, and J. M. Singer, "Searching for backbones – an efficient parallel algorithm for the traveling salesman problem," *Computer Physics Communications*, vol. 96, pp. 173–188, 1996.

[22] A. Darwiche and P. Marquis, "A knowledge compilation map," *J. Artif. Intell. Res.*, vol. 17, pp. 229–264, 2002.

[23] J. K. Slaney and T. Walsh, "Backbones in optimization and approximation," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001*, B. Nebel, Ed., 2001, pp. 254–259.