

Backdoors into Two Occurrences

Jan Johannsen

Institut für Informatik

Ludwig-Maximilians-Universität München, Germany

jan.johannsen@ifi.lmu.de

Abstract

Backdoor sets for the class CNF(2) of CNF-formulas in which every variable has at most two occurrences are studied in terms of parameterized complexity. The question whether there exists a CNF(2)-backdoor set of size k is hard for the class W[2], for both weak and strong backdoors, and in both cases it becomes fixed-parameter tractable when restricted to inputs in d -CNF for a fixed d .

Besides that, it is shown that the problem of finding weak backdoor sets is W[2]-complete, for certain tractable cases. These are the first completeness results in lower levels of the W-hierarchy for any backdoor set problems.

KEYWORDS: *Backdoor set, fixed-parameter tractability, parameterized complexity, completeness, formulas with two variable occurrences*

Submitted August 2016; revised November 2017; published February 2020

1. Introduction

Despite the theoretical hardness of the SAT problem, being the canonical NP-complete problem [1] and conjectured to not be solvable in sub-exponential time [6], state-of-the-art SAT solvers have become very efficient, and routinely solve instances arising from applications with hundreds of thousands of variables and millions of clauses. Even though there are lots of known tractable cases of SAT, i.e., classes of formulas that can be solved in polynomial time, the efficiently solvable instances arising in practice usually do not belong to these tractable classes, and thus the existence of these classes does not suffice to explain this apparent discrepancy between theory and practice.

A possibly better attempt at explaining the discrepancy is that the large, efficiently solvable instances are in some way close to a tractable case. One possible such notion of closeness is that there is a small subset of variables, such that after giving values to these variables the residual formula is in the tractable class. This concept was introduced by Crama et al. [2] and Williams et al. [13], the latter work coined the name *backdoor set* for such a set of variables.

There are several kinds of backdoor sets for each tractable case considered in the literature. A *strong backdoor set* for the class \mathcal{C} is a set of variables, such that for every setting of these variables the residual formula is in \mathcal{C} . A *weak backdoor set* for \mathcal{C} is a set of variables, such that for some setting of these variables the residual formula is in \mathcal{C} and satisfiable. There is also the auxiliary notion of a *deletion backdoor set* to be defined below.

A strong backdoor set of size k allows to solve a formula of size m in time $2^k \cdot m^{O(1)}$. This runtime bound depends on the two parameters (size of the formula m and backdoor set size k) in essentially different ways, and the proper framework to analyze such complexity

bounds is *fixed-parameter tractability* and *parameterized complexity* as introduced by Downey and Fellows [3].

Besides the class of fixed-parameter tractable problems, the theory of parameterized complexity provides a hierarchy of classes of increasingly hard problems that are probably fixed-parameter intractable, the W -hierarchy containing in particular the classes $W[t]$ for $t \in \mathbb{N}$, and a notion of reduction that allows to define hardness and completeness for these classes.

The complexity bound mentioned above thus means that given a formula together with a strong backdoor set of size k , testing for satisfiability is fixed-parameter tractable w.r.t. the parameter k . Thus, if the problem of finding a backdoor set of size k was fixed-parameter tractable w.r.t. the parameter k as well, then the size of a smallest backdoor set would be a parameter with respect to which SAT is fixed-parameter tractable. Thus, starting with the work of Nishimura et al. [11], the parameterized complexity of finding backdoor sets was determined for various tractable cases, most of the results in that direction are collected in the recent survey by Gaspers and Szeider [5].

A less well-known tractable case is the class $CNF(2)$ of CNF-formulas in which every variable has at most two occurrences. The satisfiability problem for these formulas can be solved in linear time [9], and it has been shown to be complete for deterministic logarithmic space [8].

In this work, we determine the parameterized complexity of finding backdoor sets w.r.t. the class $CNF(2)$. We show that the problem is hard for the class $W[2]$, for both weak and strong backdoor sets, and that in both cases it becomes fixed-parameter tractable when restricted to inputs in d -CNF for a fixed d .

For those tractable cases where the problem of finding backdoor sets is $W[2]$ -hard, including $CNF(2)$, the smallest parameterized complexity class known to contain the problem is $W[P]$, which lies higher up in the W -hierarchy, beyond the classes $W[t]$. For the tractable cases of 0-valid and 1-valid formulas, we show that the weak backdoor set problem is complete for $W[2]$. To the best of our knowledge, these are the first completeness results in the W -hierarchy for any weak backdoor set problems.

In order to prove this latter result, we study a related artificial problem of finding so-called *very weak* backdoor sets, whose definition differs from that of weak backdoor sets by weakening the condition that the residual formula has to be satisfiable. We show that the problem of finding very weak backdoor sets for these classes is in $W[2]$, by utilizing the characterization of the W -hierarchy in terms of first-order logic definability. The method also allows us to put the very weak backdoor set problem for other tractable cases into the class $W[2]$, including the class $CNF(2)$.

The paper is structured as follows: in Section 2 we review the necessary background on the problem SAT and its tractable cases, in particular the class $CNF(2)$, on parameterized complexity and on backdoor sets. Section 3 treats some general properties of $CNF(2)$ -backdoor sets. In Section 4 we show the results about weak $CNF(2)$ -backdoor sets, and in Section 5 those about strong $CNF(2)$ -backdoor sets. Finally Section 6 presents the mentioned upper bounds in the W -hierarchy.

2. Preliminaries

We briefly review basic notions about the propositional satisfiability problem, mainly to fix the notation.

A *literal* is a variable x or a negated variable \bar{x} . A *clause* is a disjunction $C = a_1 \vee \dots \vee a_d$ of literals a_i . The *width* $w(C)$ of C is d , the number of literals in C . We identify a clause with the set of literals occurring in it, even though for clarity we still write it as a disjunction.

A formula in *conjunctive normal form* (CNF) is a conjunction $F = C_1 \wedge \dots \wedge C_m$ of clauses, it is usually identified with the set of clauses $\{C_1, \dots, C_m\}$. A formula F in CNF is in d -CNF if every clause C in F is of width $w(C) \leq d$.

A *restriction* α is a partial assignment $\alpha : V \rightarrow \{0, 1\}$ from the set of variables V to the truth values $\{0, 1\}$. A restriction α is extended to literals by setting $\alpha(\bar{x}) := 1 - \alpha(x)$. We occasionally identify a restriction α with the set of literals it sets, i.e., $\{a; \alpha(a) = 1\}$.

For a clause C over the variables V , we define $C[\alpha = 1$ if $\alpha(a) = 1$ for some literal $a \in C$, and otherwise $C[\alpha$ is the disjunction of those literals $a \in C$ for which $\alpha(a) = 0$ does not hold, i.e., which are left unset by α . Here the empty disjunction is identified with the constant 0.

For a CNF-formula F over V , we define $F[\alpha = 0$ if $C[\alpha = 0$ for some $C \in F$, and otherwise $F[\alpha$ is the conjunction of the clauses $C[\alpha$ for those clauses $C \in F$ for which $C[\alpha = 1$ is not the case. Here the empty conjunction is identified with the constant 1.

In other words, the formula $F[\alpha$ is obtained by deleting clauses satisfied by α from F , and deleting literals falsified by α from the remaining clauses in F .

For $\epsilon = 0, 1$, we denote by $[x := \epsilon]$ the restriction setting the variable x to ϵ . This notation is also extended to literals by letting $[\bar{x} := \epsilon]$ denote $[x := (1 - \epsilon)]$.

If $F[\alpha = 1$, then we say that α satisfies F and write $\alpha \models F$. The satisfiability problem SAT is the decision problem:

Instance: Formula F in CNF.

Question: Is there a restriction α with $\alpha \models F$?

This problem SAT is the canonical NP-complete problem [1], and the strong exponential time hypothesis [6] is a widely-believed conjecture that implies that SAT is not solvable in sub-exponential time.

A clause C is *tautological*, if both x and \bar{x} occur in C for some variable x . Since tautological clauses are satisfied by all restrictions of their variables, they are irrelevant for the satisfiability of a formula they appear in. Therefore, SAT and other related problems are often restricted to formulas that do not contain any tautological clauses. Except when noted, all our results hold for problems restricted in this way.

2.1. Tractable Cases of SAT

Despite its hardness, many easy special cases of the problem SAT have been identified. A *tractable case* of SAT, sometimes also called an “island of tractability”, is a class of CNF-formulas such that

- membership $F \in \mathcal{C}$ can be decided in polynomial time,
- the satisfiability problem for formulas $F \in \mathcal{C}$ can be decided in polynomial time.

Many tractable cases of SAT have been defined and studied in the literature, some well-known examples of such classes are, e.g., the following:

- The class HORN of Horn formulas, i.e., formulas in which every clause contains at most one positive literal.
- The class CO-HORN of formulas in which every clause contains at most one negative literal.

- The class 2-CNF of formulas in which every clause is of width at most 2.
- The class 1-VAL of formulas where every clause contains at least one positive literal.
- The class 0-VAL of formulas where every clause contains at least one negative literal.

Note that formulas from the latter two classes are trivially satisfiable by the assignment setting every variable to 1 (resp., 0).

All the above tractable cases are defined by properties of the individual clauses in the formula, and by the classification result of Schaefer [12], these are the only maximal such classes of CNF formulas.

A different tractable class which is not defined by properties of clauses is the class of *cluster formulas* [7,10]. Two clauses C and C' *clash* if they contain complementary literals, i.e. if $a \in C$ and $\bar{a} \in C'$ for some literal a . A formula F is a *hitting formula* if any two different clauses in F clash. The class CLU of *cluster formulas* is the class of CNF-formulas that are variable-disjoint unions of hitting formulas.

Another less well-known tractable class is the class CNF(2) of formulas with at most two occurrences of every variable – in a way a dual to 2-CNF. It is known that satisfiability of formulas in CNF(2) can be tested in linear time [9] and in logarithmic space – in fact it is complete for the class of problems solvable in deterministic logarithmic space [8].

2.2. Parameterized Complexity

We briefly review the basic concepts of parameterized complexity as used in this work, mostly following the textbook of Flum and Grohe [4].

A parameterized problem is a decision problem P together with a parameterization, i.e., a polynomial time computable mapping that associates with every instance x of P a parameter $k = k(x) \in \mathbb{N}$. We denote by $n = n(x) \in \mathbb{N}$ the size of an instance, the number of bits required to represent x .

A parameterized problem P with parameter k is *fixed-parameter tractable* if there is an algorithm that solves P in time $f(k) \cdot O(n^d)$ for some computable function f and $d \in \mathbb{N}$. The class of parameterized problems that are fixed-parameter tractable is called FPT.

An FPT-reduction between parameterized problems P and P' with parametrizations k and k' , resp., is a function r mapping instances of P to instances of P' such that

- r is computable in FPT-time $f(k) \cdot O(n^d)$ for some computable function f and $d \in \mathbb{N}$,
- $r(x)$ is a positive instance of P' iff x is a positive instance of P ,
- there is a computable function g s.t. for every instance x of P we have $k'(r(x)) \leq g(k(x))$.

For a first-order formula $\varphi(X)$ with a free relation variable X of arity d , the parameterized problem $WD(\varphi)$ is the following:

Instance: Structure A for the language of φ , $k \in \mathbb{N}$.

Parameter: k .

Question: Is there a relation $R \subseteq A^d$ of size $|R| \leq k$ with $A \models \varphi(R)$?

A first-order formula is a Π_t -formula if it has at most t alternations of quantifiers, with the outermost quantifier being universal. A Σ_t -formula has at most t alternations of quantifiers, with the outermost one existential. The class $W[t]$ is the class of parameterized problems that are FPT-reducible to some problem $WD(\varphi)$ for a Π_t -formula $\varphi(X)$.

The parameterized problem WEIGHTED CIRCUIT SAT is:



Instance: Boolean circuit C , $k \in \mathbb{N}$.

Parameter: k .

Question: Is there a satisfying assignment $\alpha \models C$ with $|\{x; \alpha(x) = 1\}| \leq k$?

The class $W[P]$ is the class of parameterized problems that are FPT-reducible to WEIGHTED CIRCUIT SAT. Since restricted forms of WEIGHTED CIRCUIT SAT (for circuits of constant depth and *wcft* t , see [3]) are complete for the classes $W[t]$, we have that

$$\text{FPT} \subseteq W[1] \subseteq W[2] \subseteq \dots \subseteq W[t] \subseteq \dots \subseteq W[P].$$

The parameterized problem HITTING SET is the following :

Instance: Family $\{S_1, \dots, S_m\}$ of subsets $S_i \subseteq U$, $k \in \mathbb{N}$.

Parameter: k .

Question: Is there a set $H \subseteq U$ of size $|H| \leq k$ with $H \cap S_i \neq \emptyset$ for every i ?

HITTING SET is one of the canonical $W[2]$ -complete problems. To see that it is in $W[2]$, view an instance as a structure with unary predicates $U(x)$ for the elements of U , and $\text{set}(x)$ for the sets S_i , and the element relation \in between elements of U and sets. Then HITTING SET is the problem $WD(\varphi)$ for the following Π_2 -formula $\varphi(X)$ with a unary relation variable X :

$$\varphi(X) \equiv \forall x(\text{set}(x) \rightarrow \exists y(U(y) \wedge X(y) \wedge x \in y)).$$

2.3. Backdoor Sets

Let F be a CNF-formula in the variables V , and let \mathcal{C} be a tractable case of SAT. Let $X \subseteq V$ be a set of variables. The formula $F \setminus X$ is the formula obtained by deleting from the clauses in F all occurrences of literals x and \bar{x} for $x \in X$.

- A strong \mathcal{C} -backdoor set for F is a subset $X \subseteq V$ such that for every assignment $\alpha : X \rightarrow \{0, 1\}$, the formula $F \upharpoonright \alpha$ is in \mathcal{C} .
- A weak \mathcal{C} -backdoor set for F is a subset $X \subseteq V$ such that there is an assignment $\alpha : X \rightarrow \{0, 1\}$ such that the formula $F \upharpoonright \alpha$ is in \mathcal{C} and satisfiable.
- A deletion \mathcal{C} -backdoor set for F is a subset $X \subseteq V$ such that the formula $F \setminus X$ is in \mathcal{C} .

If the class \mathcal{C} is closed under subsets, then every deletion backdoor set is also a strong backdoor set, since $F \upharpoonright \alpha \subseteq F \setminus X$ for every $\alpha : X \rightarrow \{0, 1\}$. This fact is useful since deletion backdoor sets are usually easier to find than strong backdoor sets. For classes \mathcal{C} that are also closed under unions, strong and deletion backdoor sets coincide, since $F \setminus X = \bigcup_{\alpha} F \upharpoonright \alpha$ where the union is over all $\alpha : X \rightarrow \{0, 1\}$.

The parameterized problem STRONG \mathcal{C} -BACKDOOR SET is:

Instance: CNF-formula F , and $k \in \mathbb{N}$.

Parameter: k .

Question: Is there a strong \mathcal{C} -backdoor set for F of size k ?

The problems WEAK \mathcal{C} -BACKDOOR SET and DELETION \mathcal{C} -BACKDOOR SET are defined analogously.

The parameterized complexity of these problems for various tractable classes of formulas has been determined in the literature. For the classes $\mathcal{C} = \text{HORN}, \text{CO-HORN}, \text{2-CNF}, \text{1-VAL}$

and 0-VAL, the problem STRONG \mathcal{C} -BACKDOOR SET is fixed parameter tractable. Since these classes are defined by properties of individual clauses, they are closed under subsets and unions, so the problem is the same as DELETION \mathcal{C} -BACKDOOR SET, which is easily seen to be fixed parameter tractable in these cases.

The problem WEAK \mathcal{C} -BACKDOOR SET for all of these classes, on the other hand, is $W[2]$ -hard, but fixed parameter tractable when the input is restricted to formulas in d -CNF for a fixed $d \in \mathbb{N}$. For the class of cluster formulas, both problems STRONG CLU-BACKDOOR SET and WEAK CLU-BACKDOOR SET are $W[2]$ -hard, but are in FPT when restricted to inputs in d -CNF. These results, together with many more results for other tractable cases, can be found in a recent survey by Gaspers and Szeider [5].

3. Backdoors into CNF(2)

Since the class CNF(2) is obviously not closed under unions, deletion and strong backdoor sets do not necessarily coincide for this class. This is actually not the case, as the following example shows. Consider the following set of clauses:

$$(x_1 \vee x_2 \vee \bar{x}_3), \quad (\bar{x}_1 \vee x_2 \vee x_5), \quad (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_5), \quad (\bar{x}_2 \vee x_4), \quad (x_3 \vee \bar{x}_4).$$

The variables $\{x_1, x_2, x_3\}$ occur three times each, so the smallest deletion CNF(2)-backdoor set has size 3. But $\{x_1\}$ is a strong CNF(2)-backdoor set, showing that strong backdoors can be smaller than deletion backdoors for this class.

For the base class CNF(2), it matters whether formulas are represented as multisets, with multiple occurrences of the same clause allowed, or as sets, since the number of occurrences of variables is counted differently in both cases.

If formulas are represented as multisets, then the smallest deletion CNF(2)-backdoor set is exactly the set of variables with more than two occurrences, hence we trivially have:

Proposition 1. *For formulas represented as multisets, the problem DELETION CNF(2)-BACKDOOR SET can be solved in linear time.*

On the other hand, for formulas represented as sets, clauses can become equal – and hence identified – if literals are deleted or set to 0. To show that this actually makes a difference note the following proposition.

Proposition 2. *For formulas represented as sets, the problem DELETION CNF(2)-BACKDOOR SET is NP-hard.*

Proof: We reduce the well-known NP-hard problem VERTEX COVER to DELETION CNF(2)-BACKDOOR SET. For a graph $G = (V, E)$, define a formula $F = F(G)$ as follows: F has variables $x_v \in V$ for v , and for every edge $\{u, v\}$ a subformula F_e of three clauses:

$$x_u \vee x_v, \quad \bar{x}_u \vee x_v, \quad x_u \vee \bar{x}_v.$$

Now if U is a vertex cover in G , the set $X_U = \{x_u; u \in U\}$ is a deletion CNF(2)-backdoor set: after deleting the variable x_u for $u \in U$, the remaining formula consists of the unit clauses x_v and \bar{x}_v for $v \in V \setminus U$.

On the other hand, if $U \subseteq V$ is not a vertex cover, and let edge $e = \{u, v\}$ be uncovered, then the subformula F_e remains unchanged after deleting the variables in X_U , thus x_u and x_v occur at least three times each, thus X_U is not a deletion CNF(2)-backdoor set. Thus F has a deletion CNF(2)-backdoor set of size k iff G has a vertex cover of size k . \square

We will focus on the case of formulas represented as sets in the remainder of the paper.

4. Weak Backdoors

We now show the hardness of finding weak CNF(2)-backdoor sets.

Theorem 3. WEAK CNF(2)-BACKDOOR SET is W[2]-hard.

Proof: We reduce HITTING SET to WEAK CNF(2)-BACKDOOR SET. Let an instance $\mathcal{S} = \{S_1, \dots, S_m\}$ of HITTING SET be given, with $U = S_1 \cup \dots \cup S_m$. We construct a formula $F = F(\mathcal{S})$ in the variables x_s for $s \in U$ plus $z_{i,1}$ and $z_{i,2}$ for $1 \leq i \leq m$, such that F has a weak CNF(2)-backdoor set of size k iff \mathcal{S} has a hitting set of size k .

The formula F has for every set $S_i \in \mathcal{S}$ the subformula D_i consisting of the three clauses

$$\bigvee_{s \in S_i} x_s \vee \bar{z}_{i,1}, \quad z_{i,1} \vee z_{i,2}, \quad z_{i,1} \vee \bar{z}_{i,2}.$$

Let $H \subseteq U$ be a hitting set for \mathcal{S} . We show that $X_H := \{x_s; s \in H\}$ is a weak CNF(2)-backdoor set for $F(\mathcal{S})$ of the same size. Let α be the assignment with $\alpha(x_s) = 1$ for every $s \in H$. Then since H hits every set S_i , it follows that $\alpha \models \bigvee_{s \in S_i} x_s$ for every i , and hence $D_i \upharpoonright \alpha = (z_{i,1} \vee z_{i,2}) \wedge (z_{i,1} \vee \bar{z}_{i,2})$ is in CNF(2) and satisfiable for every i . Thus $F \upharpoonright \alpha$ is in CNF(2) and satisfiable.

For the other direction, let B be a weak CNF(2)-backdoor set for F , and let α be an assignment to the variables in B such that $F \upharpoonright \alpha$ is in CNF(2) and satisfiable. We first show that without loss of generality, B contains only variables x_s for $s \in U$. If B contains a variable $z_{i,b}$, then the assignment α restricted to the variables in $B \setminus \{z_{i,b}\}$ will still leave every subformula D_j for $j \neq i$ in CNF(2) and satisfiable. Thus if we let $B' = B \setminus \{z_{i,b}\} \cup \{x_s\}$ for an arbitrary $s \in S_i$, and let $\alpha'(x_s) = 1$ and α' coincide with α for all variables in $B \setminus \{z_{i,b}\}$, then $F \upharpoonright \alpha'$ is in CNF(2) and satisfiable, thus B' is a weak CNF(2)-backdoor set with $|B'| \leq |B|$.

Now we define $H_B := \{s; x_s \in B\}$, and show that H_B is a hitting set for \mathcal{S} of size $|B|$. Assume for contradiction that $S_i \cap H_B = \emptyset$. Then the formula D_i is left unchanged in $F \upharpoonright \alpha$, and therefore the variable $z_{i,1}$ has three occurrences in $F \upharpoonright \alpha$. Hence B can not be a weak CNF(2)-backdoor set. \square

In the survey [5] of Gaspers and Szeider, there is a generic construction to show the W[2]-hardness of the problem WEAK \mathcal{C} -BACKDOOR SET for a class \mathcal{C} . Our reduction is based on that construction, but it is simplified, and it has the property that the formula $F(\mathcal{S})$ only depends on \mathcal{S} and is independent from the parameter k . In other words, it is a polynomial time reduction between the underlying classical problems that does not increase the parameter. The same simplification can also be made to other applications of the generic construction in [5], e.g. for $\mathcal{C} = \text{HORN}$ and $\mathcal{C} = \text{2-CNF}$.

On the other hand, the problem of finding weak CNF(2)-backdoor sets becomes fixed-parameter tractable when restricted to inputs in 3-CNF:

Theorem 4. *The problem WEAK CNF(2)-BACKDOOR SET for 3-CNF-formulas is fixed-parameter tractable.*

Proof: We will devise a bounded search tree algorithm that, given a formula F and parameter k , will search for a restriction α of size $|\alpha| \leq k$ such that $F[\alpha]$ is in CNF(2) and satisfiable. We will call such a restriction a backdoor restriction, its domain is a weak CNF(2)-backdoor set. Obviously, a backdoor restriction exists if and only if a weak CNF(2)-backdoor set exists.

A set of three clauses C_1, C_2, C_3 in F that share a common variable x will be called an *obstruction*.

Proposition 5. *Let an obstruction C_1, C_2, C_3 in F be given. Then every backdoor restriction α for F must set some variable that occurs in C_1, C_2, C_3 .*

This holds because otherwise we have $C_i[\alpha] = C_i$ for $i = 1, 2, 3$ and therefore $F[\alpha]$ still contains the obstruction, and is thus not in CNF(2).

Once we have chosen a literal to be set, the following obvious proposition shows the correctness of the recurrence that the search tree algorithm is based on.

Proposition 6. *F has a backdoor restriction of size k that contains the assignment $[a := 1]$ iff $F[a := 1]$ has a backdoor restriction of size $k - 1$.*

These two proposition show the correctness of the following algorithm, that finds a backdoor restriction of size k in a 3-CNF-formula F if one exists.

Build a search tree of depth k , where at each node v at depth d we keep a partial assignment α_v of size $|\alpha_v| = d$. At the root we have $\alpha = \emptyset$. A node v is closed if $F[\alpha_v]$ is in CNF(2).

To extend the tree from a node v of depth $d < k$ that is not closed and labeled with α , find an obstruction C_1, C_2, C_3 in $F[\alpha]$ with the common variable x .

Now for each literal a based on a variable occurring in C_1, C_2, C_3 , add a child to v with the assignment $\alpha \cup [a := 1]$. These are at most 14 children since the clauses C_1, C_2, C_3 together contain at most 6 distinct variables besides x .

Now, for every closed leaf v labeled α , test whether $F[\alpha]$ is satisfiable. If so, α is a backdoor restriction. If there is no closed leaf, or for no closed leaf v the residual formula $F[\alpha_v]$ is satisfiable, then F does not have a backdoor restriction of size k , and hence no weak CNF(2)-backdoor set of size k .

Since every inner node has at most 14 children, the size of the search tree is 14^k , and therefore the runtime is $O(14^k n)$. \square

The algorithm generalizes in the obvious way to formulas in d -CNF for every fixed $d \in \mathbb{N}$, where the branching degree of the search tree is $2(3(d - 1) + 1) = 6d - 4$, and hence its size is $(6d - 4)^k$, which yields a runtime of $(6d - 4)^k \cdot O(n)$.

5. Strong Backdoors

Next, we show the hardness of finding strong CNF(2)-backdoor sets. Unfortunately, the proof of this result only works when the input formulas are allowed to contain tautological clauses. The complexity of the problem restricted to formulas without tautological clauses remains open.

Theorem 7. STRONG CNF(2)-BACKDOOR SET *is* W[2]-hard.

Proof: We reduce HITTING SET to STRONG CNF(2)-BACKDOOR SET. The reduction is similar to that used in the proof of Theorem 3.

Let an instance $\mathcal{S} = \{S_1, \dots, S_m\}$ of HITTING SET be given, with $U := S_1 \cup \dots \cup S_m$. We construct a formula $F = F(\mathcal{S})$ in the variables x_s for $s \in U$ plus $z_{i,1}, z_{i,2}$ for $1 \leq i \leq m$, such that F has a strong CNF(2)-backdoor set of size k iff \mathcal{S} has a hitting set of size k .

The formula F has three clauses for every set $S_i \in \mathcal{S}$, viz.

$$C_i := \bigvee_{s \in U} \bar{x}_s \vee \bigvee_{s \in S_i} x_s \vee z_{i,1}, \quad \bar{z}_{i,1} \vee z_{i,2}, \quad \bar{z}_{i,1} \vee \bar{z}_{i,2}.$$

Let $H \subseteq U$ be a hitting set for \mathcal{S} . We show that $X_H := \{x_s; s \in H\}$ is a strong CNF(2)-backdoor set for $F(\mathcal{S})$ of the same size.

Let α be an assignment to the variables in X_H . If $\alpha(x_s) = 0$ for some $s \in H$, then α satisfies $\bigvee_{s \in U} \bar{x}_s$, and hence $\alpha \models C_i$ for every i . If, on the other hand $\alpha(x_s) = 1$ for every $s \in H$, then since H is a hitting set, α satisfies $\bigvee_{s \in S_i} x_s$ and hence $\alpha \models C_i$ for every i . Thus in either case $F \upharpoonright \alpha \in \text{CNF}(2)$.

Now let B be strong CNF(2)-backdoor set for F . As in the proof of Theorem 3, we first show that without loss of generality, B contains only variables x_s for $s \in S$. If B contains a variable $z_{i,b}$, then as before we can exchange it for a variable x_s for an arbitrary $s \in S_i$.

Now as above, we define $H_B := \{s; x_s \in B\}$, and show that H_B is a hitting set for \mathcal{S} of size $|B|$. Assume for contradiction that $S_i \cap H_B = \emptyset$, and let α be the assignment with $\alpha(x_s) = 1$ for every $x_s \in B$. Then α does not satisfy any of the clauses associated with S_i , and thus the variable $z_{i,1}$ occurs three times in $F \upharpoonright \alpha$. Hence B cannot have been a strong CNF(2)-backdoor set. \square

The comment after the proof of Theorem 3 applies here as well: the reduction from HITTING SET to STRONG CNF(2)-BACKDOOR SET given is actually a polynomial time reduction that does not change the parameter.

As in the case of weak CNF(2)-backdoor sets, the problem of finding strong CNF(2)-backdoor sets becomes fixed-parameter tractable when restricted to inputs in 3-CNF.

Theorem 8. STRONG CNF(2)-BACKDOOR SET *for* 3-CNF-formulas *is* fixed-parameter tractable.

Proof: Let F be a 3-CNF-formula, and X a subset of the variables of F . The following two propositions show the correctness of the bounded search tree algorithm to be presented below.

Proposition 9. *Let $\alpha : X \rightarrow \{0, 1\}$ be a restriction. If C_1, C_2, C_3 is an obstruction in $F \upharpoonright \alpha$, then every strong CNF(2)-backdoor set for F extending X contains a variable that occurs in C_1, C_2, C_3 .*

Otherwise, if $Y \supseteq X$ is a set that is disjoint from the variables of C_1, C_2, C_3 , then for any restriction $\beta : Y \rightarrow \{0, 1\}$ that extends α the formula $F \upharpoonright \beta$ still contains the obstruction C_1, C_2, C_3 , and is therefore not in CNF(2).

Once we have chosen a variable to be included in the backdoor set, the following proposition, which holds obviously, shows the correctness of the recurrence that the search tree algorithm is based on.

Proposition 10. *F has a strong CNF(2)-backdoor set of size k that contains the variable x iff there is a set B of size $|B| \leq k - 1$ that is a strong CNF(2)-backdoor set for $F[x := 0]$ and for $F[x := 1]$.*

We build a search tree of depth k , where at each node of depth d a set of variables X_v of size $|X_v| = d$ is kept. Together with X_v we keep a set $C_v \subseteq 2^{X_v}$ of *closed* assignments, with the property that $F[\alpha \in \text{CNF}(2)$ for every $\alpha \in C_v$. A node v is *closed* if $C_v = 2^{X_v}$.

To extend the tree from a node of depth $d < k$ labeled (X, C) that is not closed, pick an assignment $\alpha : X \rightarrow \{0, 1\}$ such that $F[\alpha \notin \text{CNF}(2)$, and an obstruction, i.e., three clauses C_1, C_2, C_3 in $F[\alpha$ that share a common variable x .

For each variable y occurring in C_1, C_2, C_3 , add a child to v labeled with $X \cup \{y\}$. For each of these children, add a set C' of assignments. To determine the set C' , we first put both extensions $\alpha \cup \{[y := 0]\}$ and $\alpha \cup \{[y := 1]\}$ of every closed assignment $\alpha \in C$ into C' . Then for every open assignment $\beta \in 2^X \setminus C$ we consider the extension $\beta_0 := \beta \cup \{[y := 0]\}$, and test whether $F[\beta_0$ is in $\text{CNF}(2)$. If that is the case, we add β_0 to C' . We perform the same for the assignment $\beta_1 := \beta \cup \{[y := 1]\}$.

If a node is closed, i.e., $C = 2^X$, then X is a strong backdoor set. If on the other hand, no closed node has been found up to depth k , then by the two Propositions 9 and 10 above there is no strong backdoor set of size k .

Since the three clauses $\{C_1, C_2, C_3\}$ contain at most 7 variables, size of the search tree is bounded by 7^k . At each node we need to perform at most $2^k O(n)$ computation steps, so the runtime is bounded by $14^k O(n)$. \square

As in the case of the algorithm for weak backdoor sets, this algorithm generalizes in the obvious way to formulas in d -CNF for every fixed $d \in \mathbb{N}$, with a larger search tree size and thus a larger exponential dependence on the parameter k .

6. Upper Bounds

For most weak backdoor set problems that are not known to be in FPT, there is no exact characterization of their complexity. With the exception of a few cases which are $W[P]$ -complete, for most of the other cases it is only known that they are $W[2]$ -hard and in $W[P]$. We will now show the – to the best of our knowledge – first $W[2]$ -completeness results for weak backdoor set problems in the following theorem.

Theorem 11. *For the classes $\mathcal{C} = 1\text{-VAL}$ and 0-VAL , the problem WEAK \mathcal{C} -BACKDOOR SET is $W[2]$ -complete.*

Since finding weak backdoor sets for these classes is known to be $W[2]$ -hard [5], we only need to show that they are in $W[2]$.

Since all formulas in 1-VAL and 0-VAL are satisfiable, weak backdoor sets into these classes are the same as the *very weak* backdoor sets defined next. This is an admittedly artificial notion of backdoor set where we weaken the requirement that the residual formula is satisfiable to the condition that it is not already false.

We define a *very weak* \mathcal{C} -backdoor set for F to be a subset $X \subseteq V$ such that there is an assignment $\alpha : X \rightarrow \{0, 1\}$ such that the formula $F[\alpha$ is in \mathcal{C} and non-trivial, i.e., $F[\alpha \neq 0$.

The parameterized problem VERY WEAK \mathcal{C} -BACKDOOR SET is:

Table 1. Relations of a formula as a structure

Relation	Meaning
$\text{lit}(a)$	a is a literal
$\text{cl}(c)$	c is a clause
$\text{occ}(a, c)$	literal a occurs in clause c
$\text{comp}(a, b)$	a and b are complementary literals
$\text{pos}(a)$	a is a positive literal
$\text{neg}(a)$	a is a negative literal

Instance: CNF-formula F , and $k \in \mathbb{N}$.

Parameter: k .

Question: Is there a very weak \mathcal{C} -backdoor set for F of size k ?

For the classes 1-VAL and 0-VAL, this problem is equivalent to WEAK \mathcal{C} -BACKDOOR SET, thus to prove Theorem 11 it suffices to show it is in $W[2]$. Since our technique readily generalizes to other cases, we show that the problem of finding very weak \mathcal{C} -backdoor sets is in $W[2]$, for various tractable cases \mathcal{C} . For the other cases besides 1-VAL and 0-VAL, this might turn out to be useful in the future.

We show that these problems are in the class $W[2]$ by making use of the logical characterization of this class.

Proposition 12. *For each of the tractable classes*

$$\mathcal{C} = \text{CNF}(2), \text{HORN}, \text{CO-HORN}, \text{2-CNF}, \text{CLU}, \text{1-VAL} \text{ and } \text{0-VAL},$$

the problem VERY WEAK \mathcal{C} -BACKDOOR SET is in $W[2]$.

Proof: View a CNF-formula F as a structure whose elements are the literals of F and the clauses of F , with the relations described in Table 1.

For each of the classes \mathcal{C} in the statement of the theorem, we will define a Π_2 -formula $\varphi_{\mathcal{C}}(A)$ in this language with a free set variable A expressing that A is a backdoor restriction into the class \mathcal{C} , i.e. $F \models \varphi_{\mathcal{C}}(\alpha)$ for a restriction α iff $F \upharpoonright \alpha \in \mathcal{C}$. We also define a Π_2 -formula ntriv_A expressing that the formula F after restriction by A is nontrivial. Thus the problem VERY WEAK \mathcal{C} -BACKDOOR SET is equivalent to the problem $WD(\psi_{\mathcal{C}})$, where $\psi_{\mathcal{C}} := \text{ntriv}_A \wedge \varphi_{\mathcal{C}}(A)$. Since both formulas ntriv_A and $\varphi_{\mathcal{C}}(A)$ are Π_2 -formulas, this shows that these problems are in $W[2]$.

We start by expressing that literal a is false under A . By uniqueness of the complementary literal, this can be expressed by the following formula:

$$\text{false}_A(a) := \exists x (\text{comp}(a, x) \wedge A(x)) \equiv \forall x (\text{comp}(a, x) \rightarrow A(x)).$$

Note that this formula is a Δ_1 -formula, i.e., it is equivalent to both a Π_1 - and a Σ_1 -formula. Therefore it can be used like an atomic formula without increasing the quantifier complexity.

We then can express the condition that the formula F restricted by A is non-trivial by the Π_2 -formula

$$\text{ntriv}_A := \forall c \text{cl}(c) \rightarrow \exists a \text{lit}(a) \wedge \text{occ}(a, c) \wedge \neg \text{false}_A(a).$$

We define a Σ_1 -formula stating that the clause c is satisfied by A as:

$$\text{sat}_A(c) := \exists x(\text{occ}(x, c) \wedge A(x)).$$

The following Π_1 -formulas express that clauses c_1 and c_2 are equal after restriction by A :

$$\begin{aligned} \text{sub}_A(c_1, c_2) &:= \forall x(\text{occ}(x, c_1) \rightarrow \text{false}_A(x) \vee \text{occ}(x, c_2)), \\ \text{eq}_A(c_1, c_2) &:= \text{sub}_A(c_1, c_2) \wedge \text{sub}_A(c_2, c_1). \end{aligned}$$

We can express that the variable underlying literal a occurs in clause c by the Δ_1 -formula

$$\begin{aligned} \text{varocc}(a, c) &:= \text{occ}(a, c) \vee \exists x(\text{comp}(a, x) \wedge \text{occ}(x, c)) \\ &\equiv \text{occ}(a, c) \vee \forall x(\text{comp}(a, x) \rightarrow \text{occ}(x, c)). \end{aligned}$$

With the aid of these formulas, we can write down the formula as:

$$\begin{aligned} \forall c_1, c_2, c_3 \bigwedge_{1 \leq i \leq 3} \text{cl}(c_i) \rightarrow &\left(\bigvee_{1 \leq i \leq 3} \text{sat}_A(c_i) \vee \bigvee_{1 \leq i < j \leq 3} \text{eq}_A(c_i, c_j) \right. \\ &\left. \vee \forall a \left(\text{lit}(a) \rightarrow A(a) \vee \text{false}_A(a) \vee \bigvee_{1 \leq i \leq 3} \neg \text{varocc}(a, c_i) \right) \right). \end{aligned}$$

This formula $\varphi_{\text{CNF}(2)}(A)$ states that for any three clauses one of the following holds:

- either one of them is satisfied by A ,
- or two of them are equal under A ,
- or every variable that is not set by A does not occur in one of them.

Thus $F \models \varphi_{\text{CNF}(2)}(\alpha)$ if and only if $F \upharpoonright \alpha$ is in $\text{CNF}(2)$.

We can use the same technique to obtain upper bounds on the complexity of finding very weak backdoor sets for other tractable cases:

The following Π_2 -formula $\varphi_{2\text{-CNF}}(A)$ states that F restricted by A is a 2-CNF-formula.

$$\begin{aligned} \forall c \text{cl}(c) \rightarrow \text{sat}_A(c) \vee \forall a_1, a_2, a_3 \bigwedge_{1 \leq i \leq 3} \text{lit}(a_i) \rightarrow &\left(\bigvee_{1 \leq i < j \leq 3} a_i = a_j \right. \\ &\left. \vee \bigvee_{1 \leq i \leq 3} A(a_i) \vee \bigvee_{1 \leq i \leq 3} \text{false}_A(a_i) \vee \bigvee_{1 \leq i \leq 3} \neg \text{occ}(a_i, c) \right). \end{aligned}$$

This formula states that for every clause c that is not satisfied by A , and any three literals, one of the following holds:

- either two of the literals are equal,
- or at least one of them is set by A ,
- or at least one of them does not occur in c .

Thus $F \models \varphi_{2\text{-CNF}}(\alpha)$ if and only if $F \upharpoonright \alpha$ is in 2-CNF.

The following Π_2 -formula $\varphi_{\text{HORN}}(A)$ states that F restricted by A is a Horn formula.

$$\begin{aligned} \forall c \text{cl}(c) \rightarrow \text{sat}_A(c) \vee \forall a, a' \text{lit}(a) \wedge \text{lit}(a') \rightarrow (a = a' \vee A(a) \vee A(a')) \\ \vee \text{false}_A(a) \vee \text{false}_A(a') \\ \vee \neg \text{occ}(a, c) \vee \neg \text{occ}(a', c) \vee \text{neg}(a) \vee \text{neg}(a') \end{aligned}$$

This formula states that for every clause c unsatisfied by A , and any two distinct literals that are not set by A and occur in c , one of them is negative. The formula $\varphi_{\text{CO-HORN}}(A)$ is defined symmetrically.

A result of Nishimura et al. [10] characterizes the class of cluster formulas by excluded configurations: A formula is a cluster formula if it does not contain any of the following obstructions:

1. two clauses C and C' that *overlap*, i.e., have a variable in common, but do not clash,
2. three clauses C_1 , C_2 and C_3 such that C_1 and C_2 clash, and C_2 and C_3 clash, but C_1 and C_3 do not clash.

We define Σ_1 -formulas stating that two clauses overlap or clash as:

$$\begin{aligned} \text{overlap}(c_1, c_2) &::= \exists x \text{lit}(x) \wedge \neg A(x) \wedge \neg \text{false}_A(x) \wedge \text{varocc}(x, c_1) \wedge \text{varocc}(x, c_2), \\ \text{clash}(c_1, c_2) &::= \exists x \text{lit}(x) \wedge \neg A(x) \wedge \neg \text{false}_A(x) \wedge \text{occ}(x, c_1) \\ &\quad \wedge \exists y (\text{comp}(x, y) \wedge \text{occ}(y, c_2)). \end{aligned}$$

With the help of these formulas, we define the Π_2 -formula $\varphi_{\text{CLU}}(A)$ as $\varphi_{\text{CLU1}}(A) \wedge \varphi_{\text{CLU2}}(A)$, where $\varphi_{\text{CLU1}}(A)$ states that the formula F does not contain the first type of obstruction:

$$\begin{aligned} \forall c_1, c_2 (\text{cl}(c_1) \wedge \text{cl}(c_2) \rightarrow \text{sat}_A(c_1) \vee \text{sat}_A(c_2) \vee \text{eq}_A(c_1, c_2) \\ \vee \neg \text{overlap}(c_1, c_2) \vee \text{clash}(c_1, c_2)) \end{aligned}$$

and $\varphi_{\text{CLU2}}(A)$ states that F does not have the second type of obstruction:

$$\begin{aligned} \forall c_1, c_2, c_3 \left(\bigwedge_{1 \leq i \leq 3} \text{cl}(c_i) \rightarrow \bigvee_{1 \leq i \leq 3} \text{sat}_A(c_i) \vee \bigvee_{1 \leq i < j \leq 3} \text{eq}_A(c_i, c_j) \right. \\ \left. \vee \neg \text{clash}(c_1, c_2) \vee \neg \text{clash}(c_2, c_3) \vee \text{clash}(c_1, c_3) \right). \end{aligned}$$

Finally, we define the Π_2 -formula $\varphi_{1\text{-VAL}}(A)$ expressing that the formula F restricted by A is 1-valid. The formula $\varphi_{0\text{-VAL}}(A)$ is defined analogously.

We define the Σ_1 -formula

$$\text{cpos}(c) := \exists x \text{occ}(x, c) \wedge \neg \text{false}_A(x) \wedge \text{pos}(x)$$

expressing that clause c contains a positive literal that is not falsified by A . Thus the following Π_2 -formula

$$\varphi_{1\text{-VAL}}(A) := \forall c \text{cl}(c) \rightarrow \text{sat}_A(c) \vee \text{cpos}(c)$$

states that F restricted by A is 1-valid. □

6.1. Open Problems

We list some problems left open by this work.

- Settle the parameterized complexity of the problem STRONG CNF(2)-BACKDOOR SET in the restricted case when formulas do not contain tautological clauses, i.e., show the problem remains $W[2]$ -hard in this case.
- Determine the precise parameterized complexity of WEAK \mathcal{C} -BACKDOOR SET for tractable cases \mathcal{C} other than 1-VAL and 0-VAL, possibly using the logical approach used in this paper.
- Is the problem STRONG \mathcal{C} -BACKDOOR SET in $W[2]$ for the tractable cases $\mathcal{C} = \text{CNF}(2)$ and $\mathcal{C} = \text{CLU}$, for which we know it is $W[2]$ -hard?

Acknowledgements

I thank Stefan Szeider, Sebastian Ordyniak and Ulrich Schöpp for useful discussions about the contents of the paper, and an anonymous referee whose comments helped to improve the presentation of the paper. The research leading to the results in this paper was initiated at Dagstuhl Seminar 12471 “SAT Interactions”.

References

- [1] S.A. Cook, The complexity of theorem-proving procedures, in: *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, 1971, pp. 151–158.
- [2] Y. Crama, O. Ekin and P.L. Hammer, Variable and term removal from Boolean formulae, *Discrete Applied Mathematics* **75**(3) (1997), 217–230. doi:[10.1016/S0166-218X\(96\)00028-5](https://doi.org/10.1016/S0166-218X(96)00028-5).
- [3] R.G. Downey and M.R. Fellows, Fixed-parameter tractability and completeness, *Congressus Numerantium* **87** (1992), 161–187.
- [4] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science, Springer, 2006.
- [5] S. Gaspers and S. Szeider, Backdoors to satisfaction, in: *The Multivariate Algorithmic Revolution and Beyond*, H.L. Bodlaender, R. Downey, F.V. Fomin and D. Marx, eds, Lecture Notes in Computer Science, Vol. 7370, 2012, pp. 287–317. doi:[10.1007/978-3-642-30891-8_15](https://doi.org/10.1007/978-3-642-30891-8_15).
- [6] R. Impagliazzo, R. Paturi and F. Zane, Which problems have strongly exponential complexity?, *Journal of Computer and System Sciences* **63**(4) (2001), 512–530. doi:[10.1006/jcss.2001.1774](https://doi.org/10.1006/jcss.2001.1774).
- [7] K. Iwama, CNF-satisfiability test by counting and polynomial average time, *SIAM Journal on Computing* **18**(2) (1989), 385–391. doi:[10.1137/0218026](https://doi.org/10.1137/0218026).

- [8] J. Johannsen, Satisfiability problems complete for deterministic logarithmic space, in: *21st International Symposium on Theoretical Aspects of Computer Science (STACS 2004)*, V. Diekert and M. Habib, eds, Lecture Notes in Computer Science, Vol. 2996, 2004, pp. 317–325.
- [9] H. Kleine Büning and T. Lettmann, *Propositional Logic: Deduction and Algorithms*, Cambridge University Press, 1999.
- [10] N. Nishimura, P. Ragde and S. Szeider, Solving #SAT using vertex covers, *Acta Informatica* 44(7–8) (2007), 509–523. doi:[10.1007/s00236-007-0056-x](https://doi.org/10.1007/s00236-007-0056-x).
- [11] N. Nishimura, P. Ragde and S. Szeider, Detecting backdoor sets with respect to Horn and binary clauses, in: *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing*, 2004, pp. 96–103.
- [12] T.J. Schaefer, The complexity of satisfiability problems, in: *Proceedings of the 10th ACM Symposium on Theory of Computing*, 1978, pp. 216–226.
- [13] R. Williams, C. Gomes and B. Selman, Backdoors to typical case complexity, in: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, G. Gottlob and T. Walsh, eds, 2003, pp. 1173–1178.