

# Upper and Lower Bounds for Weak Backdoor Set Detection

Neeldhara Misra<sup>1</sup> Sebastian Ordyniak<sup>2</sup> Venkatesh Raman<sup>3</sup> Stefan Szeider<sup>4\*</sup>

<sup>1</sup> Department of Computer Science and Automation, Indian Institute of Science, Bangalore  
mail@neeldhara.com

<sup>2</sup> Faculty of Informatics, Masaryk University Brno, Czech Republic  
ordyniak@fi.muni.cz

<sup>3</sup> The Institute of Mathematical Sciences, Chennai  
vraman@imsc.res.in

<sup>4</sup> Institute of Information Systems, Vienna University of Technology, Austria  
stefan@szeider.net

## Abstract

We obtain upper and lower bounds for running times of exponential time algorithms for the detection of weak backdoor sets of 3CNF formulas, considering various base classes. These results include

- a  $4.54^k$  algorithm to detect whether there is a weak backdoor set of at most  $k$  variables for the class of Horn formulas;
- a  $2.27^k$  algorithm to detect whether there is a weak backdoor set of at most  $k$  variables for the class of Krom formulas;

both from an  $n$  variable 3CNF formula (omitting polynomial factors). These bounds improve an earlier known bound of  $6^k$ . We also prove a  $2^k$  lower bound for these problems, subject to the Strong Exponential Time Hypothesis.

## 1 Introduction

A backdoor set is a set of variables of a CNF formula such that fixing the truth values of the variables in the backdoor set moves the formula into some polynomial-time decidable class. Backdoor sets were independently introduced by Crama et al. [2] and by Williams et al. [16], the latter authors coined the term “backdoor.” The existence of a small backdoor set in a CNF formula can be considered as an indication of “hidden structure” in the formula. It is an interesting and open theoretical question, whether the worst-case time complexity of SAT decreases if the input formula has a small backdoor.

One distinguishes between various types of backdoors. Let  $\mathcal{B}$  denote the base class of formulas under consideration. A *weak  $\mathcal{B}$ -backdoor set* of a CNF formula  $F$  is a set of variables  $B$  such that there is a truth assignment  $\tau$  of the variables in  $B$  for which the formula  $F[\tau]$ , which is obtained from  $F$  by assigning the variables of  $B$  according to  $\tau$  and applying the usual simplifications, is satisfiable and  $F[\tau] \in \mathcal{B}$ . A *strong  $\mathcal{B}$ -backdoor set* of  $F$  is a set  $B$  of variables such that for *each* truth assignment  $\tau$  of the variables in  $B$ , the formula  $F[\tau]$  is in  $\mathcal{B}$ .

The challenging problem is to find a weak or strong  $\mathcal{B}$ -backdoor set of size at most  $k$  if it exists. These problems are NP-hard for all reasonable base classes. However, if  $k$  is assumed to be small, an interesting complexity landscape evolves, which can best be analyzed in the context of *parameterized complexity*, where  $k$  is considered the parameter (some basic notions of parameterized complexity will be reviewed in Section 2). This line of research was initiated by Nishimura et al. [14] who showed that for the fundamental base classes HORN and KROM, strong backdoor detection is fixed-parameter tractable, whereas weak backdoor detection is not fixed-parameter tractable (under a complexity theoretic assumption). However, if the width of the clauses of the input formula is bounded by a constant, then these hardness results do not hold any more and one achieves

---

\*Research supported by the ERC, grant reference 239962.

| $\mathcal{B}$ : | HORN         | KROM         | 0-VAL          | FOREST     | RHORN                          | QHORN                          | MATCH                          |
|-----------------|--------------|--------------|----------------|------------|--------------------------------|--------------------------------|--------------------------------|
| UB:             | $4.54^k$ [*] | $2.27^k$ [*] | $2.85^k$ [15]  | $f(k)$ [7] | $n^k$ [triv]                   | $n^k$ [triv]                   | $n^k$ [triv]                   |
| LB:             | $2^k$ [*]    | $2^k$ [*]    | $2^{o(k)}$ [*] | $2^k$ [*]  | $n^{\frac{k}{2}-\epsilon}$ [8] | $n^{\frac{k}{2}-\epsilon}$ [6] | $n^{\frac{k}{2}-\epsilon}$ [*] |

Table 1: Upper bounds (UB) and lower bounds (LB) for the time complexity of  $\text{WB}(3\text{CNF}, \mathcal{B})$  for various base classes  $\mathcal{B}$  (polynomial factors are omitted). The  $2^k$  and  $n^{\frac{k}{2}-\epsilon}$  lower bounds are subject to the Strong Exponential-Time Hypothesis, and the  $2^{o(k)}$  lower bounds are subject to the Exponential-Time Hypothesis. Results marked [\*] are obtained in this paper.

fixed-parameter tractability [8]. In order to discuss these results, we introduce the following problem template which is defined for any two classes  $\mathcal{A}, \mathcal{B}$  of CNF formulas.

$\text{WB}(\mathcal{A}, \mathcal{B})$

*Instance:* A CNF formula  $F \in \mathcal{A}$  with  $n$  variables, a non-negative integer  $k$ .

*Parameter:* The integer  $k$ .

*Question:* Does  $F$  have a weak  $\mathcal{B}$ -backdoor of size at most  $k$ ?

Thus, one could think of this problem as asking for a small weak backdoor “from  $\mathcal{A}$  to  $\mathcal{B}$ .” In this paper we focus on the special case where  $\mathcal{A} = 3\text{CNF}$ . In particular, we aim to draw a detailed complexity landscape of  $\text{WB}(3\text{CNF}, \mathcal{B})$  for various base classes, providing improved lower and upper bounds. An overview of our results in the context of known results is provided in Table 1, new results are marked with [\*]. The definitions of these classes appear in Section 2.

Gaspers and Szeider [8] showed that  $\text{WB}(3\text{CNF}, \mathcal{B})$  is fixed-parameter tractable for any base class  $\mathcal{B}$  which is defined by a property of individual clauses, such as the classes HORN, KROM, and 0-VAL. Their general algorithm provides a running time of  $6^k$  (omitting polynomial factors). We improve this to  $4.54^k$  for HORN and to  $2.27^k$  for KROM. These results fit nicely with the recent  $2.85^k$  algorithm for 0-VAL by Raman and Shankar [15].

There are base classes for which the detection of weak backdoor sets remains fixed-parameter *intractable* (in terms of  $\text{W}[2]$ -hardness), even if the input is restricted to 3CNF. In particular, the  $\text{W}$ -hardness of  $\text{WB}(3\text{CNF}, \mathcal{B})$  is known for the base class RHORN [8] and for the base class of QHORN [6]: We extend this line of results with another example. We consider the class MATCH of *matched* CNF formulas [5], which are CNF formulas  $F$  where for each clause  $C \in F$  one can select a unique variable  $x_C$  that appears in  $C$  positively or negatively, such that  $x_C \neq x_D$  for  $C \neq D$ . Since all matched formulas are satisfiable, this class is particularly well suited as a base class for weak backdoor sets. It is known that  $\text{WB}(\text{CNF}, \text{MATCH})$  is  $\text{W}[2]$ -hard, but the case  $\text{WB}(3\text{CNF}, \text{MATCH})$  has been open. We show, that  $\text{WB}(3\text{CNF}, \text{MATCH})$  is  $\text{W}[2]$ -hard as well. We further explain how the  $\text{W}[2]$ -hardness proofs can be used to get lower bounds of the form  $n^{\frac{k}{2}-\epsilon}$  under the Strong Exponential Time Hypothesis (see Section 4).

## 2 Preliminaries

**CNF Formulas and Assignments** We consider propositional formulas in conjunctive normal form (CNF) as sets of clauses, where each clause is a set of literals, i.e., a literal is either a (positive) variable or a negated variable, not containing a pair of complementary literals. We say that a variable  $x$  is positive (negative) in a clause  $C$  if  $x \in C$  ( $\bar{x} \in C$ ), and we write  $\text{var}(C)$  for the set of variables that are positive or negative in  $C$ . A *truth assignment*  $\tau$  is a mapping from a set of variables, denoted by  $\text{var}(\tau)$ , to  $\{0, 1\}$ . A truth assignment  $\tau$  *satisfies* a clause  $C$  if it sets at least one positive variable of  $C$  to 1 or at least one negative variable of  $C$  to 0. A truth assignment  $\tau$  satisfies a CNF formula  $F$  if it satisfies all clauses of  $F$ . Given a CNF formula  $F$  and a truth assignment  $\tau$ ,  $F[\tau]$  denotes the *truth assignment reduct* of  $F$  under  $\tau$ , which is the CNF formula obtained from  $F$  by first removing all clauses that are satisfied by  $\tau$  and second removing from the remaining clauses all literals  $x, \bar{x}$  with  $x \in \text{var}(\tau)$ . Note that no assignment satisfies an empty clause. The *incidence graph* of a CNF formula

$F$  is the bipartite graph whose vertices are the variables and clauses of  $F$ , and where a variable  $x$  and a clause  $C$  are adjacent if and only if  $x \in \text{var}(C)$ .

We consider the following *classes of CNF formulas*.

- 3CNF: the class of CNF formulas where each clause contains at most 3 literals.
- KROM: the class of CNF formulas where each clause contains at most 2 literals (also called 2CNF).
- HORN: the class of *Horn* formulas, i.e., CNF formulas where each clause has at most 1 positive literal.
- RHORN: the class of *renameable Horn formulas*, i.e., formulas that can be made Horn by complementing variables.
- QHORN: the class of *q-Horn formulas* [1] (RHORN, KROM  $\subseteq$  QHORN).
- 0-VAL: the class of *0-valid* CNF formulas, i.e., formulas where each clause contains at least 1 negative literal.
- FOREST: the class of *acyclic formulas* (the undirected incidence graph is acyclic).
- MATCH: the class of *matched formulas*, formulas whose incidence graph has a matching such that each clause is matched to some unique variable.

**Parameterized Complexity** Here we introduce the relevant concepts of parameterized complexity theory. For more details, we refer to text books on the topic [3, 4, 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  $(I, k)$  can be solved in time  $f(k)|I|^c$ , where  $f$  is a computable function of  $k$ , and  $c$  is a constant. FPT denotes the class of all fixed-parameter tractable problems. Hardness for parameterized complexity classes is based on *fpt-reductions*. A parameterized problem  $L$  is fpt-reducible to another parameterized problem  $L'$  if there is a mapping  $R$  from instances of  $L$  to instances of  $L'$  such that (i)  $(I, k) \in L$  if and only if  $(I', k') = R(I, k) \in L'$ , (ii)  $k' \leq g(k)$  for a computable function  $g$ , and (iii)  $R$  can be computed in time  $O(f(k)|I|^c)$  for a computable function  $f$  and a constant  $c$ . Central to the completeness theory of parameterized complexity is the hierarchy  $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots$ . Each intractability class  $\text{W}[t]$  contains all parameterized problems that can be reduced to a certain parameterized satisfiability problem under fpt-reductions.

The following problem is well-known to be  $\text{W}[2]$ -complete.

HITTING SET

*Instance:* A family  $\mathcal{S}$  of finite sets  $S_1, \dots, S_m$  and an integer  $k > 0$ .

*Parameter:* The integer  $k$ .

*Question:* Does  $\mathcal{S}$  have a hitting set of size at most  $k$ , i.e., a set  $H \subseteq \bigcup_{1 \leq i \leq m} S_i$  such that  $H \cap S_i \neq \emptyset$  for every  $1 \leq i \leq m$  and  $|H| \leq k$ ?

However, the restricted variant where all sets  $S_i$  are of size at most 3, is fixed-parameter tractable and can be solved in time  $2.270^k$ , omitting polynomial factors [13].

### 3 Upper Bounds

**Theorem 1.**  $\text{WB}(3\text{CNF}, \text{KROM})$  can be solved in time  $2.270^k$  (omitting polynomial factors).

*Proof.* Let  $F$  and  $k$  be the given 3CNF formula and non-negative integer, respectively. Let  $\mathcal{S}$  be the family of sets  $\{\text{var}(C) : C \in F, |C| = 3\}$ . Then a set  $B$  is a weak KROM-backdoor of size at most  $k$  if and only if  $B$  is a hitting set of  $\mathcal{S}$  of size at most  $k$  and  $F$  is satisfiable. Hence, we can find a weak KROM-backdoor of size at most  $k$  by finding a hitting set  $H$  of  $\mathcal{S}$  of size at most  $k$  and checking whether there is an assignment  $\tau_H$  of the variables in  $H$  such that  $F[\tau_H]$  is satisfiable. Since  $F[\tau_H] \in \text{KROM}$  the satisfiability of  $F[\tau_H]$  can be checked in polynomial-time. It follows that if we omit polynomial factors then the running time of this algorithm is the time

required to find a hitting set of  $\mathcal{S}$  of size at most  $k$ , i.e.,  $2.270^k$  [13], plus the time required to go over the at most  $2^k$  assignments of the variables in the hitting set. The correctness follows from the fact that if  $F[\tau_H]$  is satisfiable for some  $\tau_H$ , then we clearly have the desired backdoor set. On the other hand, if  $F[\tau_H]$  is not satisfiable for *any*  $\tau_H$ , then  $F$  was not satisfiable to begin with, and does not admit a weak backdoor of any size.  $\square$

**Theorem 2.**  $\text{WB}(3\text{CNF}, \text{HORN})$  can be solved in time  $(\frac{1}{2}(1 + \sqrt{65}))^k < 4.54^k$  (omitting polynomial factors).

*Proof.* Let  $F$  and  $k$  be the given 3CNF formula and non-negative integer, respectively. If  $F \in \text{HORN}$  then there is nothing to do. So suppose that  $F \notin \text{HORN}$  and let  $\text{NH}(F)$  be the set of all clauses of  $F$  that are not horn. Then  $\text{NH}(F)$  can contain the following types of clauses: (C1) clauses that contain only positive literals (and at least two of them), and (C2) clauses that contain exactly two positive literals and one negative literal.

An assignment  $\tau$  is *minimal* with respect to a formula  $F$  or to a clause  $C$  if  $F[\tau] \in \text{HORN}$  or  $\{C[\tau]\} \in \text{HORN}$ , respectively, but  $F[\tau'] \notin \text{HORN}$  or  $\{C[\tau']\} \notin \text{HORN}$  for every assignment  $\tau'$  that agrees with  $\tau$  but is defined on a strict subset of  $\text{var}(\tau)$ . Our algorithm uses the bounded search tree method to branch over all possible minimal assignments  $\tau$  that set at most  $k$  variables of  $F$  such that  $F[\tau] \in \text{HORN}$ . The algorithm then checks for each of these assignments whether  $F[\tau]$  is satisfiable (because  $F[\tau] \in \text{HORN}$  this can be done in polynomial-time). If there is at least one such assignment  $\tau$  such that  $F[\tau]$  is satisfiable, then the algorithm returns  $\text{var}(\tau)$  as a weak HORN-backdoor of  $F$  with witness  $\tau$ . Otherwise, i.e., if there is no such assignment, the algorithm outputs that  $F$  does not have a weak HORN-backdoor of size at most  $k$ .

At the root node of the search tree we set  $\tau$  to be the empty assignment. Depending on the types and structure of the clauses of the formula  $F[\tau]$ , the algorithm then branches as follows: If  $F[\tau]$  contains at least one clause of type (C1), then the algorithm branches on one of these clauses according to branching rule (R1). If  $F[\tau]$  contains at least two clauses of type (C2) that are not variable-disjoint, then the algorithm branches on such a pair according to branching rule (R2). Otherwise, i.e., if  $\text{NH}(F[\tau])$  merely consists of clauses of type (C2) which are pairwise variable-disjoint the algorithm branches according to branching rule (R3).

We will now describe the branching rules (R1)–(R3) in detail. In the following let  $\tau'$  be the assignment obtained before the current node in the search tree, and let  $x, x', y, y', z$  and  $z'$  be 6 pairwise distinct variables. Every branching rule will lead to a new assignment  $\tau$  (extending the current assignment  $\tau'$ ) where the parameter  $k$  decreases by  $|\text{var}(\tau) \setminus \text{var}(\tau')|$ .

Let  $C \in F[\tau']$  be a clause of type (C1). Then branching rule (R1) is defined as follows. If  $C = \{x, y, z\}$  then  $\{C[\tau]\} \in \text{HORN}$  if and only if  $\tau(x) = 1$  or  $\tau(y) = 1$  or  $\tau(z) = 1$  or  $\tau(x) = 0 = \tau(y)$  or  $\tau(x) = 0 = \tau(z)$ , or  $\tau(y) = 0 = \tau(z)$ . Hence, there are 3 cases for which the parameter (the number of variables set in the backdoor) decreases by 1 and 3 cases for which the parameter decreases by 2. This leads to the recurrence relation  $T(k) = 3T(k-1) + 3T(k-2) = (\frac{1}{2}(3 + \sqrt{21}))^k < 4.54^k$ . Similarly, if  $C = \{x, y\}$  then  $\{C[\tau]\} \in \text{HORN}$  if and only if  $\tau(x) = 0$  or  $\tau(x) = 1$  or  $\tau(y) = 0$  or  $\tau(y) = 1$ . Hence, there are 4 cases and in each of them the parameter decreases by 1. This leads to the recurrence function  $T(k) = 4T(k-1) = 4^k < 4.54^k$ .

Let  $C \in F[\tau']$  and  $C' \in F[\tau']$  be two distinct clauses of type (C2) that share at least one variable. Then branching rule (R2) is defined as follows.

We distinguish the following cases.

- ( $C$  and  $C'$  have exactly 1 variable  $x$  in common)

We distinguish 3 cases:

- ( $x$  is positive in  $C$  and  $C'$ )

In this case  $C = \{x, y, \bar{z}\}$  and  $C' = \{x, y', \bar{z}'\}$  and  $\{C[\tau]\}, \{C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- \*  $\tau(x) \in \{0, 1\}$ ;
- \*  $\tau(y) \in \{0, 1\}$  and  $\tau(y') \in \{0, 1\}$ ;
- \*  $\tau(y) \in \{0, 1\}$  and  $\tau(z') = 0$ ;
- \*  $\tau(z) = 0$  and  $\tau(y') \in \{0, 1\}$ ;
- \*  $\tau(z) = 0$  and  $\tau(z') = 0$ .

This leads to the recurrence function  $T(k) = 2T(k-1) + 9T(k-2) = (1 + \sqrt{10})^k < 4.54^k$ .

– (*x is negative in C and C'*)

In this case  $C = \{\bar{x}, y, z\}$  and  $C' = \{\bar{x}, y', z'\}$  and  $\{C[\tau], \{C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- \*  $\tau(x) = 0$ ;
- \*  $\tau(y) \in \{0, 1\}$  and  $\tau(y') \in \{0, 1\}$ ;
- \*  $\tau(y) \in \{0, 1\}$  and  $\tau(z') \in \{0, 1\}$ ;
- \*  $\tau(z) = \{0, 1\}$  and  $\tau(y') \in \{0, 1\}$ ;
- \*  $\tau(z) = \{0, 1\}$  and  $\tau(z') = \{0, 1\}$ .

This leads to the following recurrence function:  $T(k) = T(k-1) + 16T(k-2) = (\frac{1}{2}(1 + \sqrt{65}))^k < 4.54^k$ .

– (*x is negative in C and positive in C'*)

In this case  $C = \{\bar{x}, y, z\}$  and  $C' = \{x, y', \bar{z}'\}$  and  $\{C[\tau], \{C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- \*  $\tau(x) = 0$ ;
- \*  $\tau(x) = 1$  and  $\tau(y) \in \{0, 1\}$ ;
- \*  $\tau(x) = 1$  and  $\tau(z) \in \{0, 1\}$ ;
- \*  $\tau(y) \in \{0, 1\}$  and  $\tau(y') \in \{0, 1\}$ ;
- \*  $\tau(y) \in \{0, 1\}$  and  $\tau(z') = 0$ ;
- \*  $\tau(z) \in \{0, 1\}$  and  $\tau(y') \in \{0, 1\}$ ;
- \*  $\tau(z) \in \{0, 1\}$  and  $\tau(z') = 0$ .

This leads to the recurrence function  $T(k) = T(k-1) + 14T(k-2) < 4.54^k$ .

- (*C and C' have exactly 2 variables x and y in common*)

We distinguish 3 cases:

– (*x and y are positive in C and C'*)

In this case  $C = \{x, y, \bar{z}\}$  and  $C' = \{x, y, \bar{z}'\}$  and  $\{C[\tau], \{C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- \*  $\tau(x) \in \{0, 1\}$ ;
- \*  $\tau(y) \in \{0, 1\}$ ;
- \*  $\tau(z) = 0$  and  $\tau(z') = 0$ ;

This leads to the recurrence function  $T(k) = 4T(k-1) + T(k-2) = (2 + \sqrt{5})^k < 4.54^k$ .

– (*x is positive in C and C' and y is negative in C and C'*)

In this case  $C = \{x, \bar{y}, z\}$  and  $C' = \{x, \bar{y}, z'\}$  and  $\{C[\tau], \{C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- \*  $\tau(x) \in \{0, 1\}$ ;
- \*  $\tau(y) = 0$ ;
- \*  $\tau(z) \in \{0, 1\}$  and  $\tau(z') \in \{0, 1\}$ .

This leads to the recurrence function  $T(k) = 3T(k-1) + 4T(k-2) < 4.54^k$ .

– (*x is positive in C and negative in C' and y is negative in C and positive in C'*)

In this case  $C = \{x, \bar{y}, z\}$  and  $C' = \{\bar{x}, y, z'\}$  and  $\{C[\tau], \{C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- \*  $\tau(x) = 0$ ;
- \*  $\tau(y) = 0$ ;
- \*  $\tau(x) = 1$  and  $\tau(y) = 1$ ;
- \*  $\tau(x) = 1$  and  $\tau(z') \in \{0, 1\}$ ;

- \*  $\tau(y) = 1$  and  $\tau(z) \in \{0, 1\}$ ;
- \*  $\tau(z) \in \{0, 1\}$  and  $\tau(z') \in \{0, 1\}$ .

This leads to the recurrence function:  $T(k) = 2T(k-1) + 9T(k-2) = (1 + \sqrt{10})^k < 4.54^k$ .

- ( $C$  and  $C'$  have exactly 3 variables  $x, y$  and  $z$  in common)

In this case  $C = \{x, y, \bar{z}\}$  and  $C' = \{x, \bar{y}, z\}$  and  $\{C[\tau], C'[\tau]\} \in \text{HORN}$  if and only if one of the following holds:

- $\tau(x) \in \{0, 1\}$ ;
- $\tau(y) = 0$ ;
- $\tau(z) = 0$ ;
- $\tau(y) = 1$  and  $\tau(z) = 1$ .

This leads to the recurrence function  $T(k) = 4T(k-1) + T(k-2) = (2 + \sqrt{5})^k < 4.54^k$ .

Taking the maximum over the above cases we obtain the recurrence function  $T(k) = T(k-1) + 16T(k-2) = (\frac{1}{2}(1 + \sqrt{65}))^k < 4.54^k$  for branching rule (R2).

Recall that after applying branching rule (R2) exhaustively all pairs of clauses of type (C2) are pairwise variable-disjoint. We will describe branching rule (R3), which makes use of this fact. Let  $C = \{x, y, \bar{z}\}$ . We set either  $\tau(x) = 0$  or  $\tau(x) = 1$ . This leads to the recurrence function:  $T(k) = 2T(k-1) = 2^k < 4.54^k$ . Note that in contrast to the branching rules (R1) and (R2) the branching rule (R3) is not exhaustive. Indeed for every clause  $C = \{x, y, \bar{z}\}$  of type (C2) there are 5 possible minimal assignments  $\tau$  such that  $\{C[\tau]\} \in \text{HORN}$ , i.e., the assignments  $\tau(x) = 0, \tau(x) = 1, \tau(y) = 0, \tau(y) = 1$ , and  $\tau(z) = 0$ . Because each of these assignments  $\tau$  sets only 1 variable this would lead to a recurrence function  $T(k) = 5T(k-1) = 5^k$  and hence  $T(k) > 4.54^k$ . It follows that in contrast to the branching rules R1 and R2 where we could exhaustively branch over all possible minimal assignments, this can not be done for clauses of type (C2). However, because branching rule (R2) ensures that the remaining clauses of type (C2) are pairwise variable-disjoint it turns out that this is indeed not necessary (see Claim 1).

This concludes the description of our algorithm. The running time of the algorithm is the maximum branching factor over the cases described above, i.e.,  $(\frac{1}{2}(1 + \sqrt{65}))^k < 4.54^k$  as required. To see that the algorithm is correct we need to show that it outputs an assignment  $\tau$  if and only if the set  $\text{var}(\tau)$  is a weak HORN-backdoor of  $F$  of size at most  $k$ . Because the branching rules R1 and R2 branch exhaustively over all minimal assignments  $\tau$  such that the corresponding clause(s) are reduced to Horn clauses, it only remains to show the correctness of branching rule R3. This is done by the following claim.

**Claim 1.** *Let  $F$  be a 3CNF formula,  $P$  be a set of pairwise variable-disjoint clauses of type (C2) such that  $F \setminus P \in \text{HORN}$ . Furthermore, let  $L$  be a set of variables that consists of one positively occurring variable from each of the clauses in  $P$ . Then  $F$  has a weak HORN-backdoor of size at most  $|P|$  if and only if  $L$  is a weak HORN-backdoor of  $F$ .*

To prove the claim, suppose that  $F$  has a weak HORN-backdoor  $B$  of size at most  $k$ . Hence, there is an assignment  $\tau^s$  that satisfies  $F$ . Let  $\tau_L$  be an assignment of the variables of  $F$  that agrees with  $\tau^s$ . Then  $F[\tau_L] \in \text{HORN}$  (this actually holds for every assignment of the variables in  $L$ ) and  $\tau_L$  can be extended to a satisfying assignment of  $F$ . Hence,  $L$  is a weak HORN-backdoor of  $F$  of size at most  $|L| = |P|$  with witness  $\tau_L$  as required. The reverse direction follows from the fact that  $|L| = |P|$ . Hence the claim is established and the theorem follows.  $\square$

## 4 Lower Bounds

For our lower bounds we use the *Exponential Time Hypothesis* (ETH) and the *Strong Exponential Time Hypothesis* (SETH), introduced by Impagliazzo et al. [9, 10], which state the following:

**ETH:** There is no algorithm that decides the satisfiability of a 3CNF formula with  $n$  variables in time  $2^{o(n)}$ , omitting polynomial factors.

**SETH:** There is no algorithm that decides the satisfiability of a CNF formula with  $n$  variables in time  $(2 - \epsilon)^n$ , omitting polynomial factors.

An *implication chain* is a CNF formula of the form  $\{\{x_0\}, \{\bar{x}_0, x_1\}, \{\bar{x}_1, x_2\}, \dots, \{\bar{x}_{n-1}, x_n\}, \{\bar{x}_n\}\}$ ,  $n \geq 1$  where the first  $\{x_0\}$  and the last clause  $\{\bar{x}_n\}$  can be missing. Let CHAINS denote the class of formulas that are variable-disjoint unions of implication chains.

**Theorem 3.** *Let  $\mathcal{B}$  be a base class that contains CHAINS. Then  $\text{WB}(3\text{CNF}, \mathcal{B})$  cannot be solved in time  $(2 - \epsilon)^k$  (omitting polynomial factors) unless SETH fails.*

*Proof.* We show that an  $(2 - \epsilon)^k$  algorithm for WEAK CHAINS-BACKDOOR SET DETECTION implies an  $(2 - \epsilon)^n$  algorithm for SAT contradicting our assumption. Let  $F$  be a CNF formula with  $n$  variables. We will transform  $F$  to a 3CNF formula  $F_3$  such that  $F$  is satisfiable if and only if  $F_3$  has a weak CHAINS-backdoor of size at most  $n$ . We obtain  $F_3$  from  $F$  using a commonly known transformation that transforms an arbitrary CNF formula into a 3CNF formula that is satisfiability equivalent with the original formula. In particular, we obtain the formula  $F_3$  from  $F$  by replacing every clause  $C = \{x_1, \dots, x_l\}$  where  $l > 3$  with the clauses  $\{x_1, x_2, y_1\}, \{y_1, x_3, y_2\}, \dots, \{y_{l-3}, x_l\}$ , where  $y_1, \dots, y_{l-3}$  are new variables. This completes the construction of  $F_3$ . It is well-known that  $F$  is satisfiable if and only if the  $F_3$  is satisfiable.

It remains to show that  $F$  is satisfiable if and only if  $F_3$  has a weak CHAINS-backdoor of size at most  $n$ . Suppose that  $F$  is satisfiable and let  $\tau$  be a satisfying assignment of  $F$ . We claim that the variables of  $F$  form a weak CHAINS-backdoor set of size  $n$  of  $F_3$  with witness  $\tau$ . It is easily verified that  $F_3[\tau] \in \text{CHAINS}$  and furthermore because  $F$  is satisfiable so is  $F_3$ .

The reverse direction follows immediately from the fact that  $F_3$  is satisfiable (because it has a weak CHAINS-backdoor set) and the fact that the formulas  $F$  and  $F_3$  are satisfiability equivalent by construction.  $\square$

As the classes HORN, KROM, and FOREST contain CHAINS, we have the following.

**Corollary 1.** *Let  $\mathcal{B} \in \{\text{HORN}, \text{KROM}, \text{FOREST}\}$ . Then the problem  $\text{WB}(3\text{CNF}, \mathcal{B})$  cannot be solved in time  $(2 - \epsilon)^k$  (omitting polynomial factors) unless SETH fails.*

Interestingly, in the case of KROM the above result even holds if a hitting set for all clauses containing 3 literals is given as the input.

**Theorem 4.**  *$\text{WB}(\text{KROM}, 0\text{-VAL})$ , and hence also  $\text{WB}(3\text{CNF}, 0\text{-VAL})$ , cannot be solved in time  $2^{o(k)}$  (omitting polynomial factors) unless ETH fails.*

*Proof.* To show the theorem we need to introduce the following problem.

VERTEX COVER

*Instance:* A graph  $G$  and a non-negative integer  $k$ .

*Parameter:* The integer  $k$ .

*Question:* Does  $G$  have a vertex cover of size at most  $k$ , i.e., is there a subset  $C$  of the vertices of  $G$  of cardinality at most  $k$  such that  $C$  contains at least one endpoint of every edge of  $G$ ?

Given a graph  $G$  and an integer  $k$  we construct in linear-time a KROM formula  $F$  with  $|V(G)|$  variables such that  $G$  has a vertex cover of size at most  $k$  if and only if  $F$  has a weak 0-VAL-backdoor of size at most  $k$ . Because VERTEX COVER cannot be solved in time  $2^{o(k)}$  (omitting polynomial factors) unless ETH fails [11][Theorem 3.3] this shows the theorem. The variables of  $F$  are the vertices of  $G$  and  $F$  consists of 1 clause  $\{u, v\}$  for every edge  $\{u, v\} \in E(G)$ . This completes the construction of  $F$ . It remains to show that  $G$  has a vertex cover of size at most  $k$  if and only if  $F$  has a weak 0-VAL-backdoor of size at most  $k$ .

Suppose that  $G$  has a vertex cover  $C$  of size at most  $k$ . We claim that  $C$  is a weak 0-VAL-backdoor of  $F$  of size at most  $k$ . Let  $\tau$  be the assignment of the variables in  $C$  that sets all variables in  $C$  to 1. Then  $F[\tau] = \emptyset \in 0\text{-VAL}$  because  $C$  is a vertex cover of  $G$ .

For the reverse direction suppose that  $B$  is a weak 0-VAL-backdoor of  $F$  of size at most  $k$  and  $\tau$  is a witnessing assignment of the variables in  $B$ . We claim that  $B$  is a vertex cover of  $G$  of size at most  $k$ . Suppose not, then there is an edge  $\{u, v\} \in E(G)$  such that  $B \cap \{u, v\} = \emptyset$ . Consequently,  $F[\tau]$  contains the clause  $\{u, v\}$  which is not 0-valid contradicting our assumption that  $F[\tau] \in 0\text{-VAL}$ .  $\square$

Let  $\mathcal{B}$  be a base class. We say that a polynomial-time algorithm  $\mathcal{A}$  is a *canonical HS reduction for  $\mathcal{B}$*  if  $\mathcal{A}$  takes as input an instance  $(\mathcal{S}, k)$  of HITTING SET over  $n$  elements and outputs an instance  $(F, k)$  of  $\text{WB}(3\text{CNF}, \mathcal{B})$  such that: (a)  $F$  has at most  $n^2$  variables, and (b)  $\mathcal{S}$  has a hitting set of size at most  $k$  if and only if  $F$  has a weak  $\mathcal{B}$ -backdoor of size at most  $k$ .

**Lemma 1.** *Let  $\mathcal{B}$  be a base class. If there is a canonical HS reduction for  $\mathcal{B}$ , then the following holds:*

1.  $\text{WB}(3\text{CNF}, \mathcal{B})$  is  $\text{W}[2]$ -hard, and
2. there is no algorithm that solves  $\text{WB}(3\text{CNF}, \mathcal{B})$  in time  $O(n^{\frac{k}{2}-\epsilon})$  unless SETH fails.

*Proof.* Because HITTING SET is  $\text{W}[2]$ -complete and a canonical HS reduction is also an fpt-reduction, the first statement of the theorem follows. To see the second statement we first note that it is shown in [11][Theorem 5.8] that the DOMINATING SET problem cannot be solved in time  $O(n^{k-\epsilon})$  for any  $\epsilon > 0$  unless SETH fails (here  $n$  is the number of vertices of the input graph and  $k$  is the parameter). Using the standard reduction from DOMINATING SET to HITTING SET it follows that HITTING SET cannot be solved in time  $O(n^{k-\epsilon})$  for any  $\epsilon > 0$ , where  $n$  is the number of elements of the hitting set instance and  $k$  is the parameter. Now suppose that for some base class  $\mathcal{B}$  it holds that  $\text{WB}(3\text{CNF}, \mathcal{B})$  can be solved in time  $n^{\frac{k}{2}-\epsilon}$  and  $\mathcal{B}$  has a canonical HS reduction. Let  $(\mathcal{S}, k)$  be an instance of HITTING SET with  $n_h$  elements. We use the canonical HS reduction to obtain an instance  $(F, k)$  of  $\text{WB}(3\text{CNF}, \mathcal{B})$  where  $F$  has  $n_b$  variables such that  $n_b \leq n_h^2$ . We now use the algorithm for  $\text{WB}(3\text{CNF}, \mathcal{B})$  to solve HITTING SET in time  $O(n_b^{\frac{k}{2}-\epsilon}) \leq O(n_h^{k-\frac{\epsilon}{2}})$  which contradicts our assumption that there is no such algorithm for HITTING SET.  $\square$

**Lemma 2.** *There is a canonical HS reduction for MATCH.*

*Proof.* Let  $(\mathcal{S}, k)$  be an instance of HITTING SET with  $\mathcal{S} = \{S_1, \dots, S_m\}$  and  $V = \bigcup_{i=1}^m S_i = \{x_1, \dots, x_n\}$ . We write  $S_i = \{x_i^1, \dots, x_i^{q_i}\}$ , where  $q_i = |S_i|$ . We construct in linear-time a 3CNF formula  $F$  with  $|V| + \sum_{1 \leq i \leq m} (q_i - 1) \leq n^2$  variables such that  $\mathcal{S}$  has a hitting set of size at most  $k$  if and only if  $F$  has a weak MATCH-backdoor of size at most  $k$ . This shows the theorem.

The variables of  $F$  consist of the elements of  $V$  and additional variables  $y_i^j$  for every  $1 \leq i \leq m$  and  $1 \leq j < q_i$ . We let  $F = \bigcup_{i=0}^m F_i$  where the formulas  $F_i$  are defined as follows.

$F_0$  consists of  $n$  binary clauses  $\{\bar{x}_1, x_2\}, \{\bar{x}_2, x_3\}, \dots, \{\bar{x}_{n-1}, x_n\}, \{\bar{x}_n, x_1\}$ . For  $i > 0$ ,  $F_i$  consists of the clauses  $\{y_i^1, x_i^1\}, \{\bar{y}_i^1, y_i^2, x_i^2\}, \{\bar{y}_i^2, y_i^3, x_i^3\}, \dots, \{\bar{y}_i^{q_i-2}, y_i^{q_i-1}, x_i^{q_i-1}\}, \{\bar{y}_i^{q_i-1}, x_i^{q_i}\}$ . This completes the construction of  $F$ .

We claim that  $\mathcal{S}$  has a hitting set of size at most  $k$  if and only if  $F$  has a weak MATCH-backdoor set of size at most  $k$ . Let  $H \subseteq V$  be a hitting set of  $\mathcal{S}$ . We show that  $H$  is a weak MATCH-backdoor set of  $F$ . Let  $\tau \in 2^H$  be the truth assignment that sets all variables in  $H$  to 1 and consider  $F[\tau]$ . Every clause  $C \in F_0[\tau]$  contains a unique negative literal for some variable  $x \in V$ ; we match  $C$  to  $x$ . Now consider  $F_i[\tau]$  for  $i > 0$ . We observe that  $F_i[\tau] \subseteq F_i$ . Since  $H$  is a hitting set,  $\tau$  satisfies at least one clause  $C_H$  in  $F_i$  such that  $H \cap C_H = \{x_i^j\}$ . Hence, for every clause  $C \in F_i[\tau]$  either there is an  $1 \leq \ell < j$  such that  $y_i^\ell \in C$  or there is an  $j \leq \ell < q_i$  such that  $\bar{y}_i^\ell \in C$ . This constitutes a matching for the clauses in  $F_i[\tau]$ .

For the reverse direction, let  $B$  be a weak MATCH-backdoor set of  $F$  and let  $\tau$  be an assignment of the variables in  $B$  witnessing this. We claim that the set  $B$  is a hitting set of  $\mathcal{S}$ . Assume to the contrary that for some  $1 \leq i \leq m$  we have  $S_i \cap B = \emptyset$ . Because  $F_0[\tau]$  has at least  $|V \setminus B|$  clauses and each of these clauses contains only variables in  $V \setminus B$  it follows that every clause in  $F_0[\tau]$  must be matched to a variable in  $V \setminus B$ . Consequently, for every  $1 \leq i \leq m$  the clauses in  $F_i[\tau]$  need to be matched to the variables  $y_i^1, \dots, y_i^{q_i-1}$ . Because  $S_i \cap B = \emptyset$  it follows that the number of variables in  $\{y_i^1, \dots, y_i^{q_i-1}\} \setminus B$  is 1 less than the number of clauses of  $F_i[\tau]$ . Consequently,  $F_i[\tau] \notin \text{MATCH}$  contradicting our assumption that  $B$  is a weak MATCH-backdoor of  $F$ . Hence the lemma follows.  $\square$

We observe that the known  $\text{W}[2]$ -hardness proofs for  $\text{WB}(3\text{CNF}, \text{RHORN})$  and  $\text{WB}(3\text{CNF}, \text{QHORN})$  [8, 6] are in fact canonical HS reductions. Hence, together with Lemmas 1 and 2 we arrive at the following result.

**Theorem 5.** *Let  $\mathcal{B} \in \{\text{MATCH}, \text{RHORN}, \text{QHORN}\}$ . Then  $\text{WB}(3\text{CNF}, \mathcal{B})$  is  $\text{W}[2]$ -hard and cannot be solved in time  $O(n^{\frac{k}{2}-\epsilon})$  for any  $\epsilon > 0$  unless SETH fails.*



## 5 Conclusion

We have initiated a systematic study of determining the complexity of finding weak backdoors of small size of 3CNF formulas for various base classes. We have given improved algorithms for some of the base classes through the bounded search techniques.

Our lower bounds are among the very few known bounds based on the (Strong) Exponential-Time Hypotheses for parameterized problems where the parameter is the solution size (as opposed to some measure of structure in the input like treewidth).

Closing the gaps between upper and lower bounds of the problems we considered in this paper, and studying  $WB(\mathcal{A}, \mathcal{B})$  for other classes  $\mathcal{B}$  than 3CNF are interesting directions for further research.

## References

- [1] E. Boros, P. L. Hammer, and X. Sun. Recognition of  $q$ -Horn formulae in linear time. *Discr. Appl. Math.*, 55(1):1–13, 1994.
- [2] Y. Crama, O. Ekin, and P. L. Hammer. Variable and term removal from Boolean formulae. *Discr. Appl. Math.*, 75(3):217–230, 1997.
- [3] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.
- [4] J. Flum and M. Grohe. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Springer Verlag, Berlin, 2006.
- [5] J. Franco and A. Van Gelder. A perspective on certain polynomial time solvable classes of satisfiability. *Discr. Appl. Math.*, 125:177–214, 2003.
- [6] S. Gaspers, S. Ordyniak, M. S. Ramanujan, S. Saurabh, and S. Szeider. Backdoors to  $q$ -Horn. In *The 30th Symposium on Theoretical Aspects of Computer Science will be held in Kiel, Feb 27Mar 2, 2013*, 2013. In Press.
- [7] S. Gaspers and S. Szeider. Backdoors to acyclic sat. In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 363–374. Springer Verlag, 2012.
- [8] S. Gaspers and S. Szeider. Backdoors to satisfaction. In H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 287–317. Springer Verlag, 2012.
- [9] R. Impagliazzo and R. Paturi. On the complexity of  $k$ -SAT. *J. of Computer and System Sciences*, 62(2):367–375, 2001.
- [10] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. of Computer and System Sciences*, 63(4):512–530, 2001.
- [11] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the European Association for Theoretical Computer Science*, 105:41–72, 2011.
- [12] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.
- [13] R. Niedermeier and P. Rossmanith. An efficient fixed-parameter algorithm for 3-hitting set. *J. Discrete Algorithms*, 1(1):89–102, 2003.

- [14] N. Nishimura, P. Ragde, and S. Szeider. Detecting backdoor sets with respect to Horn and binary clauses. In *Proceedings of SAT 2004 (Seventh International Conference on Theory and Applications of Satisfiability Testing, 10–13 May, 2004, Vancouver, BC, Canada)*, pages 96–103, 2004.
- [15] V. Raman and B. S. Shankar. Improved fixed-parameter algorithm for the minimum weight 3-sat problem. In S. K. Ghosh and T. Tokuyama, editors, *WALCOM: Algorithms and Computation, 7th International Workshop, WALCOM 2013, Kharagpur, India, February 14-16, 2013. Proceedings*, volume 7748 of *Lecture Notes in Computer Science*, pages 265–273. Springer Verlag, 2013.
- [16] R. Williams, C. Gomes, and B. Selman. Backdoors to typical case complexity. In G. Gottlob and T. Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI 2003*, pages 1173–1178. Morgan Kaufmann, 2003.