Backdoors into Heterogeneous Classes of SAT and CSP

Serge Gaspers  
UNSW and NICTA  
Sydney, Australia  
sergeg@cse.unsw.edu.au

Neeldhara Misra  
Indian Institute of Science  
Bangalore, India  
mail@neeldhara.com

Sebastian Ordyniak  
Masaryk Univ.  
Brno, Czech Republic  
sordyniak@gmail.com

Stefan Szeider  
Vienna Univ. of Technology  
Vienna, Austria  
stefan@szeider.net

Stanislav Živný  
Univ. of Oxford  
Oxford, UK  
standa@cs.ox.ac.uk

Abstract

Backdoor sets represent clever reasoning shortcuts through the search space for SAT and CSP. By instantiating the backdoor variables one reduces the given instance to several easy instances that belong to a tractable class. The overall time needed to solve the instance is exponential in the size of the backdoor set, hence it is a challenging problem to find a small backdoor set if one exists; over the last years this problem has been subject of intensive research.

In this paper we extend the classical notion of a strong backdoor set by allowing that different instantiations of the backdoor variables result in instances that belong to different base classes; the union of the base classes forms a heterogeneous base class. Backdoor sets to heterogeneous base classes can be much smaller than backdoor sets to homogeneous ones, hence they are much more desirable but possibly harder to find.

We draw a detailed complexity landscape for the problem of detecting strong backdoor sets into heterogeneous base classes for SAT and CSP. We provide algorithms that establish fixed-parameter tractability under natural parameterizations, and we contrast the tractability results with hardness results that pinpoint the theoretical limits.

Our results apply to the current state-of-the-art of tractable classes of CSP and SAT that are definable by restricting the constraint language.

Introduction

Backdoors are small sets of variables of a SAT or CSP instance that represent “clever reasoning shortcuts” through the search space. Backdoor sets were originally introduced by Williams, Gomes, and Selman (2003a, 2003b) as a tool for the analysis of decision heuristics in propositional satisfiability. Since then, backdoor sets have been widely used in the areas of propositional satisfiability (Williams, Gomes, and Selman 2005a), [Ruan, Kautz, and Horvitz 2004] [Dilkina, Gomes, and Sabharwal 2007] [Samer and Szeider 2009b] [Kottler, Kaufmann, and Sinz 2008] [Dilkina, Gomes, and Sabharwal 2009] [Gaspers and Szeider 2013], and also for material discovery (LeBras et al. 2013), abductive reasoning (Pfandler, Rümmele, and Szeider 2013), answer set programming (Fichte and Szeider 2013), argumentation (Dvorák, Ordyniak, and Szeider 2012), and quantified Boolean formulas (Samer and Szeider 2009a). A backdoor set is defined with respect to some fixed base class for which the computational problem under consideration is polynomial-time tractable (alternatively, it can be defined with respect to a polynomial-time subsolver). The size of the backdoor set can be seen as a distance measure that indicates how far the instance is from the target class. One distinguishes between strong and weak backdoor sets; the latter applies only to satisfiable instances, and in this paper we shall focus on the former. Once a strong backdoor set of size $k$ is identified, one can decide the satisfiability of the instance by deciding the satisfiability of at most $d^k$ “easy” instances that belong to the tractable base class, where $d$ denotes the size of the domain for the variables; for SAT we have $d = 2$. Each of the easy instances is obtained by one of the $d^k$ possible instantiations of the $k$ variables in the backdoor set. Hence, the satisfiability check is fixed-parameter tractable for the combined parameter backdoor size and domain size $(k + d)$.

The fixed-parameter tractability of using the backdoor set for deciding satisfiability triggers the question of whether finding a backdoor set of size at most $k$ is also fixed-parameter tractable. In particular, for every base class $C$ one can ask whether the detection of a strong backdoor set into $C$ of size at most $k$ is fixed-parameter tractable for parameter $k$ (possibly in combination with restrictions on the input or other parameters). A systematic study of the parameterized complexity of backdoor set detection was initiated by [Nishimura, Ragde, and Szeider (2004) for SAT, who showed that the detection of strong backdoor sets into the classes HORN and 2CNF (of Horn formulas and 2CNF formulas, respectively) is fixed-parameter tractable. Since then, the parameterized complexity of backdoor set detection has become an active research topic as outlined in a recent survey (Gaspers and Szeider 2012).

In this work, we provide two significant extensions to the exciting research on fixed-parameter tractable backdoor set detection. First, we extend the classical notion of a strong backdoor set by allowing that different instantiations of backdoor variables result in instances that belong to different base classes; the union of the base classes forms a heterogeneous base class, in contrast to the usual homogeneous base classes. Second, we extend the scope of backdoor set detection from SAT to CSP, considering target classes that are defined by tractable constraint languages in terms of closure properties under polymorphisms.
Heterogeneous Base Classes Consider the following SAT instance $F_n = \{C, D_1, \ldots, D_n\}$ where $C = (x \vee \neg a_1 \vee \cdots \vee \neg a_n)$ and $D_i = (\neg x \vee b_i \vee c_i)$. It is easy to see that any strong backdoor set into HORN needs to contain at least one of the variables $b_i$ or $c_i$ from each clause $D_i$, hence such a backdoor set must be of size $\Omega(n)$; on the other hand, any strong backdoor set into 2CNF must contain at least $(n - 2)$ variables from the clause $C$; hence such a backdoor must be of size $\Omega(n)$ as well. However, $F_n[x = \text{false}]$ is Horn, and $F_n[x = \text{true}]$ is a 2CNF, hence the singleton $\{x\}$ constitutes a strong backdoor set into the “heterogeneous” base class HORN $\cup$ 2CNF. This example shows that by considering heterogeneous base classes we can access structural properties of instances that are not accessible by backdoor sets into homogeneous base classes. Identifying a base class with a class of instances that are solvable by a polynomial-time subsolver, one can consider a heterogeneous base class as a “portfolio subsolver,” where for each instance the best suitable subsolver from the portfolio is chosen.

A natural question at this point is whether the fixed-parameter tractability results for the detection of strong backdoor sets into individual base classes HORN and 2CNF can be extended to the more powerful heterogeneous base class HORN $\cup$ 2CNF. Our first main result (Theorem1) answers this question affirmatively. The same clearly holds by taking the class of anti-Horn formulas instead of Horn. However, somewhat surprisingly, we get W[2]-hardness (i.e., fixed-parameter intractability) for other combinations of base classes (Theorem2). If we also bound the clause length by the parameter, we obtain fixed-parameter tractability in all considered cases (Theorem3).

CSP Backdoor Sets The identification of tractable classes of CSP instances has been subject of extensive studies. A prominent line of research, initiated by Schaefer (1978) in his seminal paper on Boolean CSP, is to identify tractable classes by restricting the relations that may appear in constraints to a prescribed set, a constraint language. Today, many constraint languages have been identified that give rise to tractable classes of CSPs (Pearson and Jeavons 1997; Bulatov and Dalmau 2006): typically such languages are defined in terms of certain closure properties, which ensure that the relations are closed under pointwise application of certain polymorphisms of the domain. For instance, consider a CSP instance whose relations are closed under a constant function $f(x) = d$ for some $d \in D$ (such a function is a polymorphism). Then note that every relation is either empty or forced to contain the tuple $\langle d, d, \ldots, d \rangle$. Thus, given a particular instance, we may either declare it unsatisfiable (if it contains a constraint over the empty relation), or satisfy it trivially by setting every variable to $d$. Further examples of polymorphisms for which closure properties yield tractable CSP are min, max, majority, affine, and Mal’cev polymorphisms (Jeavons, Cohen, and Gyssens 1997; Bulatov and Dalmau 2006).

We study the problem of finding strong backdoor sets into tractable classes of CSP instances defined by certain polymorphisms. Our main result for CSP backdoors (Theorem4) establishes fixed-parameter tractability for a wide range of such base classes. In particular, we show that the detection of strong backdoor sets is fixed-parameter tractable for the combined parameter backdoor size, domain size, and the maximum arity of constraints. In fact, this result entails heterogeneous base classes, as different instantiations of the backdoor variables can lead to reduced instances that are closed under different polymorphisms (even polymorphisms of different type). We complement our main result with hardness results that show that we lose fixed-parameter tractability when we omit either domain size or the maximum arity of constraints from the parameter (Theorems5 and 6). Hence Theorem4 is tight in a certain sense.

Preliminaries

SAT A literal is a propositional variable $x$ or a negated variable $\neg x$. We also use the notation $x = x^1$ and $\neg x = x^0$. A clause is a finite set of literals that does not contain a complementary pair $x$ and $\neg x$. A propositional formula in conjunctive normal form, or CNF formula for short, is a set of clauses. For a clause $C$ we write $\text{var}(C) = \{x : x \in C \text{ or } \neg x \in C\}$, and for a CNF formula $F$ we write $\text{var}(F) = \bigcup_{C \in F} \text{var}(C)$.

For a set $X$ of propositional variables we denote by $2^X$ the set of all mappings $\tau : X \to \{0, 1\}$, the truth assignments on $X$. We denote by $X$ the set of literals corresponding to the negated variables of $X$. For $\tau \in 2^X$ we let $\text{true}(\tau) = \{x^\tau(x) : x \in X\}$ and $\text{false}(\tau) = \{x^{\neg \tau}(x) : x \in X\}$ be the sets of literals set by $\tau$ to 1 and 0, respectively. Given a CNF formula $F$ and a truth assignment $\tau \in 2^X$ we define $F[\tau] = \{C \in F : C \cap \text{true}(\tau) = \emptyset\}$. If $\tau \in 2^X$ and $\epsilon = \tau(x)$, we simply write $F[x = \epsilon]$ instead of $F[\tau]$.

A CNF formula $F$ is satisfiable if there is some $\tau \in 2^{\text{var}(F)}$ with $F[\tau] = \emptyset$, otherwise $F$ is unsatisfiable.

CSP Let $D$ be a set and $n$ and $n'$ be natural numbers. An $n$-ary relation on $D$ is a subset of $D^n$. For a tuple $t \in D^n$, we denote by $t[i], i$ the $i$-th entry of $t$, where $1 \leq i \leq n$. For two tuples $t \in D^n$ and $t' \in D^{n'}$, we denote by $t \circ t'$, the concatenation of $t$ and $t'$.

A constraint satisfaction problem (CSP) $I$ is a triple $(V, D, C)$, where $V$ is a finite set of variables over a finite set (domain) $D$, and $C$ is a set of constraints. A constraint $c \in C$ consists of a scope, denoted by $\text{var}(c)$, which is an ordered list of a subset of $V$, and a relation, denoted by $\text{arc}(c)$, which is a $|\text{var}(c)|$-ary relation on $D$. To simplify notation, we sometimes treat ordered lists without repetitions, such as the scope of a constraint, like sets. For a variable $v \in V(c)$ and a tuple $t \in \text{arc}(c)$, we denote by $t[v], i$ the $i$-th entry of $t$, where $i$ is the position of $v$ in $\text{arc}(c)$. For a CSP $I = (V, D, C)$ we sometimes denote by $\text{var}(I), \text{arc}(I), C(I)$, and $\delta(I)$, its set of variables $V$, its domain $D$, its set of constraints $C$, and the maximum arity of any constraint of $I$, respectively.

Let $V' \subseteq V$ and $\tau : V' \to D$. For a constraint $c \in C$, we denote by $c[\tau], c[\tau]$, the constraint whose scope is $\text{var}(c) \setminus V'$ and whose relation contains all $\text{var}(c[\tau])$-ary tuples $t'$ such that there is a $\text{var}(c)$-ary tuple $t' \in R(c)$ with $t'[v] = t'[v]$ for every $v \in \text{var}(c[\tau])$ and $t'[v] = \tau(v)$ for every $v \in V'$. We denote by $I[\tau]$ the CSP instance with variables $V \setminus V'$, domain
Backdoors are defined relative to some fixed class \( C \) of instances of the problem under consideration (i.e., SAT or CSP). One usually assumes that the problem is tractable for instances from \( C \), as well as that the recognition of \( C \) is tractable.

In the context of SAT, we define a strong \( C \)-backdoor set of a CNF formula \( F \) to be a set of \( B \) variables such that \( F[τ] ∈ C \) for each \( τ ∈ 2^B \). If we know a strong \( C \)-backdoor set of \( F \), we can decide the satisfiability of \( F \) by checking the satisfiability of \( 2^k \) “easy” formulas \( F[τ] \) that belong to \( C \). Thus SAT decision is fixed-parameter tractable in the size \( k \) of the backdoor. Similarly, in the context of CSP, we define a strong \( C \)-backdoor set of a CSP instance \( I = (V, D, C) \) as a set of \( B \) variables such that \( I[τ] ∈ C \) for every \( τ : B → D \). We also call a strong \( C \)-backdoor a strong backdoor set into \( C \). If we know a strong \( C \)-backdoor set of \( I \), we can reduce the satisfiability of \( I \) to the satisfiability of \( d^k \) CSP instances in \( C \) where \( d = |D| \). Thus deciding the satisfiability of a CSP instance is fixed-parameter tractable in the combined parameter \( d + k \).

The challenging problem is—for SAT and for CSP—to find a strong \( C \)-backdoor set of size at most \( k \) if one exists.

For each class \( C \) of SAT or CSP instances, we define the following problem.

---

**Strong \( C \)-Backdoor Detection**

**Input:** A SAT or CSP instance \( I \) and a nonnegative integer \( k \).

**Question:** Does \( I \) have a strong \( C \)-backdoor set of size at most \( k \)?

---

**Parameterized Complexity**

We provide basic definitions of parameterized complexity; for an in-depth treatment we refer to the recent monograph [Downey and Fellows 2013]. A problem is parameterized if each problem instance \( I \) is associated with a nonnegative integer \( k \), the parameter. A parameterized problem is fixed-parameter tractable (or FPT, for short) if there is an algorithm, \( A \), a constant \( c \), and a computable function \( f \), such that \( A \) solves instances of input size \( n \) and parameter \( k \) in time \( f(k)n^c \). Fixed-parameter tractability extends the conventional notion of polynomial-time tractability, the latter being the special case where \( f \) is a polynomial. The so-called Weft-hierarchy \( W[1] ⊆ W[2] ⊆ \ldots \) contains classes of parameterized decision problems that are presumed to be larger than FPT. It is believed that problems that are hard for any of the classes in the Weft-hierarchy are not fixed-parameter tractable. The classes are closed under \( fpt \)-reductions that are fixed-parameter tractable many-one reductions, which map an instance \( x \) with parameter \( k \) of one problem to a decision-equivalent instance \( x' \) with parameter \( k' \) of another problem, where \( k' ≤ f(k) \) for some computable function \( f \).

For instance, the following problem is well-known to be \( W[2] \)-complete (Downey and Fellows 2013).

---

**Backdoor Detection for SAT**

Schaefer’s base classes [Schafer 1978] give rise to classes of CNF formulas defined in terms of syntactical properties of clauses. A clause is

- **Horn** if it contains at most one positive literal,
- **Anti-Horn** if it contains at most one negative literal,
- **2CNF** if it contains at most two literals,
- **0-valid** if it contains at least one negative literal, and
- **I-valid** if it contains at least one positive literal.

A CNF formula is Horn, Anti-Horn, etc. if it contains only Horn, Anti-Horn, etc. clauses. We denote the respective classes of CNF formulas by HORN, HORN\(^-\), 2CNF, 0-VAL, and 1-VAL.

**Strong \( C \)-Backdoor Detection** is polynomial for 0-VAL and 1-VAL, and FPT for the remaining Schaefer classes [Nishimura, Ragde, and Szeider 2004] (Gaspers and Szeider 2012). These FPT algorithms are based on constant-size obstruction sets. For a clause \( c \), we say that a set \( X \) of variables is a \( C \)-obstruction for \( c \) if \( (X ∪ \overline{X}) \cap c \) \( \notin C \) and \( (X ∪ \overline{X}) \cap c \setminus \{l\} \) \( \notin C \) for all literals \( l \in c \). A HORN-obstruction contains two variables occurring positively in the clause, a HORN\(^-\)-obstruction contains two variables occurring negatively in the clause, and a 2CNF-obstruction contains three variables occurring positively or negatively in the clause. It is well-known that, for \( C \in \{\text{HORN}, \text{HORN}^-, \text{2CNF}\} \), every strong \( C \)-backdoor set contains a variable from each \( C \)-obstruction for each clause in the formula (Nishimura, Ragde, and Szeider 2004) (Gaspers and Szeider 2012). Our next algorithm also uses these obstruction sets, but the heterogeneity of the base class makes it significantly more challenging to design FPT algorithms. Nevertheless, we obtain FPT algorithms for the base classes HORN U 2CNF and HORN\(^-\) U 2CNF.

**Theorem 1.** Strong \( \text{HORN} \cup \text{2CNF} \)-Backdoor Detection and Strong \( \text{HORN}^- \cup \text{2CNF} \)-Backdoor Detection are fixed-parameter tractable for parameter \( k \).

**Proof.** We describe the algorithm \( M \) for the base class \( \text{HORN} \cup \text{2CNF} \). The case \( \text{HORN}^- \cup \text{2CNF} \) is symmetric. Algorithm \( M \) is a recursive search tree algorithm, which is executed with parameters \( (F, k, C) \), where \( C \) is initialized to \{HORN, 2CNF\}. It returns the set of all (inclusion-wise) minimal \( C \)-backdoor sets of size at most \( k \), where \( C = \bigcup C \).

If there exists a class \( C' \in \mathcal{C} \) such that \( F \in C' \), then return \( \emptyset \) since the only minimal \( C' \)-backdoor set is the empty set.

Otherwise, if \( k = 0 \), then return \( \emptyset \) since \( F \) has no backdoor set of size at most 0.

Otherwise, if \( F \) has a clause \( c \) such that there exists no \( C' \in \mathcal{C} \) with \( \{c\} \in C' \), then let \( X \) be a set of at most 3 variables such that for each \( C' \in \mathcal{C} \), the set \( X \) contains a \( C' \)-obstruction for \( c \). Only 3 variables are necessary since
a 2CNF-obstruction can be chosen that is a superset of a HORN-obstruction. Observe that each strong \( C' \)-backdoor set contains a variable from \( X \). This is because for every set of variables avoiding \( X \), there exists an assignment that does not satisfy the clause \( c \), and for this assignment, \( c \) still contains a \( C' \)-obstruction for all \( C' \in \mathcal{C} \). The algorithm computes the set \( R \) which is the union of the sets \( \{ \{ x \} \cup_{r \in \mathcal{C}} \{ r \cup_{x \in X} \} \leq k-1, r_x \in \mathbf{M}(F[x = 1], k-1, \mathcal{C}), r_{x \in X} \in \mathbf{M}(F[x = 0], k-1, \mathcal{C}) \} \) over all \( x \in X \) and it obtains \( R^* \) by minimalizing \( R \), i.e., by removing sets from \( R \) that are proper supersets of other sets in \( R \). It returns \( R^* \). This recursion explores all possibilities for the strong \( C' \)-backdoor sets to contain a variable from \( X \). For one such variable \( x \in X \), every strong \( C' \)-backdoor set for \( F \) containing \( x \) is obtained by taking the union of \{ \( x \) \}, a strong \( C' \)-backdoor set for \( F[x = 1] \), and a strong \( C' \)-backdoor set for \( F[x = 0] \).

Otherwise, we have that \( |\mathcal{C}| = 2 \), each clause in \( F \) belongs to a class \( C' \in \mathcal{C} \), and for each class \( C' \in \mathcal{C} \) there is at least one clause that belongs to this class and not the other. Let \( c = \{ u, v \} \in F \) be a 2CNF clause that is not HORN. The algorithm computes the set \( R \) which is the union of the sets \( \{ \{ x \} \cup_{r \in \mathcal{C}} \{ r \cup_{x \in X} \} \leq k-1, r_x \in \mathbf{M}(F[x = 1], k-1, \mathcal{C}), r_{x \in X} \in \mathbf{M}(F[x = 0], k-1, \mathcal{C}) \} \) over all \( x \in c \) and the set \( \mathbf{M}(F, k, \{ 2CNF \}) \), and it obtains \( R^* \) by minimalizing \( R \). It returns \( R^* \). The correctness of this branching follows because strong \( C \)-backdoor sets that contain neither \( u \) nor \( v \) are strong 2CNF-backdoor sets.

The number of leaves of the search tree is upper bounded by \( O(6^k) \). The number of backdoor sets computed at each leaf is upper bounded by the quadratic recurrence defined by \( N(k) = 3 \cdot (N(k-1))^2 \) for \( k > 0 \) and \( N(0) = 1 \), which solves to \( 3^2 = 1 \). Therefore, the running time of the algorithm is \( 2O(2^k)nO(1) \).

The previous algorithm crucially depends on the fact that 2CNF clauses have bounded length. However, for combinations of base classes such as HORN \( \cup \) HORN', we obtain intractability results. Even more surprisingly, if we combine the very simple base classes 0-VAL and 1-VAL, we obtain a intractability result. Even more surprisingly, if we combine the very simple base classes 0-VAL and 1-VAL, we obtain a intractability result.

**Theorem 2.** For every \( C \in \{ \text{HORN}, 0-\text{VAL} \} \) and \( C' \in \{ \text{HORN'}, 1-\text{VAL} \} \), the problem STRONG \( C \cup \text{C}' \)-BACKDOOR DETECTION is \( W[2]-\text{hard} \).

**Proof.** We give a parameterized reduction from the \( W[2]-\text{complete} \) HITTING SET problem. Given an instance \((\mathcal{F}, \mathcal{U}, k)\) for HITTING SET, construct a formula \( F \) as follows. The variables of \( F \) are \( \mathcal{U} \cup \{v_F : S \in \mathcal{F}\} \cup \{v_{\mathcal{U}}\} \). For each set \( S \in \mathcal{F} \), there is one clause \( c_S = S \cup \{v_F\} \). There is also one clause \( c_{\mathcal{U}} = \{\neg u : u \in \mathcal{U}\} \) \cup \{v_{\mathcal{U}}\} \). This completes the description of the reduction.

We claim that \( F \) has a hitting set of size at most \( k \) if and only if the formula \( F \) has a strong \( C \cup \text{C}' \)-backdoor set of size at most \( k \). Suppose \( X \subseteq \mathcal{U} \), \( |X| \leq k \), is a hitting set. To show that \( X \) is also a strong \( C \cup \text{C}' \)-backdoor set, consider any assignment \( \tau \in 2^X \). If \( \tau(x) = 0 \) for some \( x \in X \), then \( \tau \) satisfies the clause \( c_{\mathcal{U}} \). Thus, \( F[\tau] \in \mathcal{C}' \) since each clause in \( F[\tau] \) contains no negative literal and at least one positive literal. If \( \tau(x) = 1 \) for all \( x \in X \), then all clauses \( c_S, S \in \mathcal{F} \), are satisfied by \( \tau \) since \( X \) is a hitting set. The only remaining clause is HORN and 0-VAL since it has no positive literal and at least one negative literal. For the other direction, suppose that \( X \) is a strong \( C \cup \text{C}' \)-backdoor set of size at most \( k \). Obtain \( X' \) from \( X \) by replacing each \( v_S \in X \) by some variable from \( S \), and if \( v_U \in X \), by replacing \( v_U \) by an arbitrary variable from \( \mathcal{U} \). The set \( X' \) is also a strong \( C \cup \text{C}' \)-backdoor set of size at most \( k \). Therefore, the assignment \( \tau \in 2^{X'} \) with \( \tau(x) = 1 \) for all \( x \in X' \) must satisfy all clauses \( c_S, S \in F \). Thus, \( X' \) is a hitting set for \( F \) of size at most \( k \).

It is crucial for these hardness proofs that clauses have unbounded length. Indeed, if clause-lengths are bounded or if we add the maximum clause length to the parameter, then strong backdoor detection becomes \( \text{FPT} \) for any combination of Schaefer classes.

**Theorem 3.** Let \( C \) be a base class consisting of the union of some of the classes HORN, HORN', 2CNF, 0-VAL, 1-VAL. Then, STRONG \( C \)-BACKDOOR DETECTION is fixed-parameter tractable for the combined parameter \( k + r \), where \( r \) is the maximum clause length of the input formula.

The proof of the above theorem resembles the proof of Theorem 1 and is omitted due to space constraints.

We close this section by noting that backdoor sets with empty clause detection, as proposed by (Dilkina, Gomes, and Sabharwal 2007) can be considered as backdoor sets into the heterogeneous base class obtained by the union of a homogeneous base class \( C \) and the class of all formulas that contain the empty clause. The detection of strong backdoor sets with empty clause detection is not fixed-parameter tractable for many natural base classes, including Horn and 2CNF (Szeider 2008).

**Base Classes via Closure Properties**

In this section, we provide a very general framework that will give rise to a wide range of heterogeneous base classes for CSP.

Given a \( k \)-ary relation \( R \) over some domain \( D \) and a function \( \phi : D^k \to D \), we say that \( R \) is closed under \( \phi \), if for all collections of \( n \) tuples \( t_1, \ldots, t_n \) from \( R \), the tuple \( (\phi(t_1[1], \ldots, t_n[1]), \ldots, \phi(t_1[k], \ldots, t_n[k])) \) belongs to \( R \). The function \( \phi \) is also said to be a polymorphism of \( R \). We denote by \( \text{Pol}(R) \) the set of all polynomials \( \phi \) such that \( R \) is closed under \( \phi \).

Let \( I = (V, D, C) \) be a CSP instance and \( c \in C \). We write \( \text{Pol}(c) \) for the set \( \text{Pol}(R(c)) \) and we write \( \text{Pol}(I) \) for the set \( \bigcap_{c \in C} \text{Pol}(c) \). We say that \( I \) is closed under a polymorphism \( \phi \) if \( \phi \in \text{Pol}(I) \).

- A polymorphism \( \phi : D \to D \) is constant if there is a \( d \in D \) such that for every \( d' \in D \), it holds that \( \phi(d') = d \);
- A polymorphism \( \phi : D^k \to D \) is idempotent if for every \( d \in D \) it holds that \( \phi(d, \ldots, d) = d \);
- A polymorphism \( \phi : D^2 \to D \) is a \text{minmax} polymorphism if there is an ordering of the elements of \( D \) such that for every \( d, d' \in D \), it holds that \( \phi(d, d') = \phi(d', d) = \min\{d, d'\} \) or \( \phi(d, d') = \phi(d', d) = \max\{d, d'\} \), respectively;
- A polymorphism \( \phi : D^3 \to D \) is a \text{majority} polymorphism if for every \( d, d' \in D \) it holds that \( \phi(d, d', d) = \phi(d, d', d) = \phi(d', d, d) = \phi(d', d, d) = d \);
A polymorphism \( \phi : D^3 \rightarrow D \) is an affine (or minority) polymorphism if for every \( d, d' \in D \) it holds that 
\[
\phi(d, d, d') = \phi(d, d', d) = \phi(d', d, d) = d'.
\]
A polymorphism \( \phi : D^3 \rightarrow D \) is a Mal'cev polymorphism if for every \( d, d' \in D \) it holds that 
\[
\phi(d, d, d') = \phi(d', d, d) = d'.
\]
We say a polymorphism \( \phi \) is tractable if every CSP instance closed under \( \phi \) can be solved in polynomial time. It is known that every constant, min/max, majority, affine, and Mal’cev polymorphism is tractable [Jeavons, Cohen, and Gyssens 1997; Bulatov and Dalmau 2006]. We denote by \( \text{VAL}, \text{MIN}, \text{MAX}, \text{MAJ}, \text{AFF}, \) and \( \text{MAL} \) the class of CSP instances \( I \) for which \( \text{Pol}(I) \) contains a constant, a min, a max, a majority, an affine, or a Mal’cev polymorphism, respectively.

Let \( \mathbb{P}(\phi) \) a predicate for polymorphisms \( \phi \). We call \( \mathbb{P}(\phi) \) a nice polymorphism property if the following conditions hold.

- There is a constant \( c_\phi \) such that for all finite domains \( D \), all polymorphisms \( \phi \) over \( D \) with property \( \mathbb{P} \) are of arity at most \( c_\phi \).
- Given a polymorphism \( \phi \), one can check in polynomial time whether \( \mathbb{P}(\phi) \) holds.
- Every polymorphism with property \( \mathbb{P} \) is tractable.

Every nice polymorphism property \( \mathbb{P} \) gives rise to a natural base class \( C_\mathbb{P} \) consisting of all CSP-instances \( I \) such that \( \text{Pol}(I) \) contains some polymorphism \( \phi \) with \( \mathbb{P}(\phi) \). Thus \( \text{VAL}, \text{MIN}, \text{MAX}, \text{MAJ}, \text{AFF}, \) and \( \text{MAL} \) are the classes \( C_\mathbb{P} \) for \( \mathbb{P} \in \{\text{constant}, \text{min}, \text{max}, \text{majority}, \text{affine}, \text{Mal’cev}\} \), respectively.

In terms of the above definitions we can state the results of Jeavons, Cohen, and Gyssens [1997] and Bulatov and Dalmau [2006] as follows.

**Proposition 1.** Constant, min, max, majority, affine, and Mal’cev are nice polymorphism properties.

In the next sections we will study the problems \( \text{STRONG } C_\mathbb{P}-\text{BACKDOOR DETECTION} \) for nice polymorphism properties \( \mathbb{P} \).

**Tractability of Backdoor Detection for CSP**

In this section we will show that \( \text{STRONG } C_\mathbb{P}-\text{BACKDOOR DETECTION} \) parameterized by the size of the backdoor set, the size of the domain, and the maximum arity of the CSP instance are fixed-parameter tractable for any nice property \( \mathbb{P} \).

**Theorem 4.** Let \( \mathbb{P} \) be a nice polymorphism property. Then \( \text{STRONG } C_\mathbb{P}-\text{BACKDOOR DETECTION} \) is fixed-parameter tractable for the combined parameter size of the backdoor set, size of the domain, and the maximum arity of the given CSP instance.

**Proof.** Let \( \mathbb{P} \) be a nice property, and let \( \langle I, k \rangle \) with \( I = \langle V, D, C \rangle \) be an instance of \( \text{STRONG } C_\mathbb{P}-\text{BACKDOOR DETECTION} \). Let \( P \) be the set of all polymorphisms on \( D \) that have property \( \mathbb{P} \). Then, \( P \) can be constructed in \( \text{fpt-time} \) with respect to the size of the domain, because there are at most \( |D|^{|D|^3} \) \( c_\mathbb{P} \)-ary polymorphisms on \( D \) and for each of them we can test in polynomial time, \( |D|^{O(c_\mathbb{P})} \), whether it satisfies property \( \mathbb{P} \). The algorithm uses a depth-bounded search tree approach to find a strong \( C_\mathbb{P} \)-backdoor set of size at most \( k \).

We construct a search tree \( T \), for which every node is labeled by a set \( B \) of at most \( k \) variables of \( V \). Additionally, every leaf node has a second label, which is either \( \text{YES} \) or \( \text{NO} \). \( T \) is defined inductively as follows. The root of \( T \) is labeled by the empty set. Furthermore, if \( t \) is a node of \( T \), whose first label is \( B \), then the children of \( t \) in \( T \) are obtained as follows. If for every assignment \( \tau : B \rightarrow D \) there is a polymorphism \( \phi \in P \) such that \( I[\tau] \) is closed under \( \phi \), then \( B \) is a strong \( P \)-backdoor set of size at most \( k \), and hence \( t \) becomes a leaf node, whose second label is \( \text{YES} \). Otherwise, i.e., if there is an assignment \( \tau : B \rightarrow D \) such that \( I[\tau] \) is not closed under any polymorphism \( \phi \in P \), we consider two cases: (1) \( |B| = k \), then \( t \) becomes a leaf node, whose second label is \( \text{NO} \), and (2) \( |B| < k \), then for every polymorphism \( \phi \in P \) and every variable \( v \) in the scope of some constraint \( c \in C[\tau] \) that is not closed under \( \phi \), \( t \) has a child whose first label is \( B \cup \{v\} \).

If \( T \) has a leaf node, whose second label is \( \text{YES} \), then the algorithm returns the first label of that leaf node. Otherwise the algorithm return \( \text{NO} \). This completes the description of the algorithm.

We now show the correctness of the algorithm. First, suppose the search tree \( T \) built by the algorithm has a leaf node \( t \) whose second label is \( \text{YES} \). Here, the algorithm returns the first label, say \( B \) of \( t \). By definition, we obtain that \( |B| \leq k \) and for every assignment \( \tau : B \rightarrow D \), it holds that \( I[\tau] \) is closed under some polymorphism in \( P \), as required.

Now consider the case where the algorithm returns \( \text{NO} \). We need to show that there is no set \( B \) of at most \( k \) variables of \( I \) such that \( \text{Pol}(I[\tau]) \cap P \neq \emptyset \) for every assignment \( \tau \) of the variables of \( B \). Assume, for the sake of contradiction that such a set \( B \) exists.

Observe that if \( T \) has a leaf node \( t \) whose first label is a set \( B' \) with \( B' \subseteq B \), then the second label of \( t \) must be \( \text{YES} \). This is because, either \( |B'| < k \) in which case the second label of \( t \) must be \( \text{YES} \), or \( |B'| = k \) in which case \( B' = B \) and by the definition of \( B \) it follows that the second label of \( t \) must be \( \text{YES} \).

It hence remains to show that \( T \) has a leaf node whose first label is a set \( B' \) with \( B' \subseteq B \). This will complete the proof about the correctness of the algorithm. We will show a slightly stronger statement, namely, that for every natural number \( \ell \), either \( T \) has a leaf whose first label is contained in \( B \) or \( T \) has an inner node of distance exactly \( \ell \) from the root whose first label is contained in \( B \). We show the latter by induction on \( \ell \).

The claim obviously holds for \( \ell = 0 \). So assume that \( T \) contains a node \( t \) at distance \( \ell \) from the root of \( T \) whose first label, say \( B' \), is a subset of \( B \). If \( t \) is a leaf node of \( T \), then the claim is shown. Otherwise, there is an assignment \( \tau : B' \rightarrow D \) such that \( I[\tau] \) is not closed under any polymorphism from \( P \). Let \( \tau^* : B \rightarrow D \) be any assignment of the variables in \( B \) that agrees with \( \tau \) on the variables in \( B' \) and let \( \phi \in P \) be such that \( I[\tau^*] \) is closed under \( \phi \). Because \( B \) is a strong \( P \)-backdoor set, the polymorphism \( \phi \) clearly exists. By definition of the search tree \( T \), \( t \) has a child \( t' \) for every variable \( v \) in the scope of some constraint \( c \in C[\tau] \) that is not closed under \( \phi \). We claim that \( V(c) \cap B \neq \emptyset \) and hence \( t' \) has a child, whose first label is a subset of \( B \), as required.
Indeed, suppose not. Then $c \in C[\tau^*]$ a contradiction to our assumption that $I[\tau^*]$ is closed under $\phi$. This concludes our proof concerning the correctness of the algorithm.

The running time of the algorithm is obtained as follows. Let $T$ be a search tree obtained by the algorithm. Then the running time of the depth-bounded search tree algorithm is $O(|V(T)|)$ times the maximum time that is spend on any node of $T$. Since the number of children of any node of $T$ is bounded by $|P|\delta(I)$ (recall that $\delta(I)$ denotes the maximum arity of any constraint of $I$) and the longest path from the root of $T$ to some leaf of $T$ is bounded by $k+1$, we obtain that $|V(T)| \leq O(|P|\delta(I))^{k+1}$.

Furthermore, the time required for any node $t$ of $T$ is at most $O(|D|^k \text{comp}_\text{rest}(I, \tau)|C(I[\tau])||P|\text{check}_\text{poly}(c, \phi))$, where $\text{comp}_\text{rest}(I, \tau)$ is the time required to compute $I[\tau]$ for some assignment $\tau$ of at most $k$ variables and $\text{check}_\text{poly}(c, \phi)$ is the time required to check whether a constraint $c$ of $I[\tau]$ preserves the polymorphism $\phi \in P$. Observe that $\text{comp}_\text{rest}(I, \tau)$ and $|C(I[\tau])|$ are polynomial in the input size. The same holds for $\text{check}_\text{poly}(c, \phi)$, because $\phi$ is a $c_p$-ary polymorphism. Now, the total running time required by the algorithm is the time required to compute the set $P$ plus the time required to compute $T$. Putting everything together, we obtain $O((|P|\delta(I))^{k+1}|D|^n|O(1)) = O((|D|^{k+1}\delta(I)|D|)|O(1)))$, as the total running time of the algorithm, where $n$ denotes the input size of the CSP instance. This shows that STRONG $C_p$-BACKDOOR DETECTION is fixed-parameter tractable parameterized by $k$, $\delta(I)$, and $|D|$.

**Corollary 1.** Let $C$ be a base class consisting of the union of some of the classes MIN, MAX, MAJ, AFF, and MAL. Then STRONG $C$-BACKDOOR DETECTION is fixed-parameter tractable for the combined parameter size of the backdoor set, size of the domain, and the maximum arity of the given CSP instance.

Recently, Bessiere et al. (2013) presented a different approach to CSP backdoors. However, it can be shown that the backdoor sets that arise from their approach are less general than strong backdoor sets, i.e., every partition backdoor set of a CSP instance is also a strong backdoor set, moreover, there are classes of CSPs for which strong backdoor sets are of size $1$, but the size of a smallest partition backdoor set grows with the number of variables of the CSP instance.

**Hardness of Backdoor Detection for CSP**

In this section we show our parameterized hardness results for STRONG $C_p$-BACKDOOR DETECTION. In particular, we show that STRONG $C_p$-BACKDOOR DETECTION is $W[2]$-hard parameterized by the size of the backdoor set even for CSP instances of Boolean domain and for CSP instances with arity two. We start by showing hardness for CSP instances over the Boolean domain.

Let $\phi : D^n \to D$ be an $n$-ary polymorphism over $D$ and $r$ a natural number. We say a sequence of $r$-ary tuples $(t_1, \ldots, t_n)$ is an obstruction for $\phi$ if $\phi(t_1, \ldots, t_n) \notin \{t_1, \ldots, t_n\}$. We say that a polymorphism is obstructable if it has an obstruction. Observe that all tractable polymorphisms are also obstructable because every CSP instance is closed under any polymorphism that is not obstructable. For a sequence $S$ of tuples, we denote by $D(S)$, the set of pairwise distinct tuples in $S$. We call an obstruction $\langle t_1, \ldots, t_n \rangle$ of $\phi$ minimal if $\delta(|D(\{t_1, \ldots, t_n\})|)$ is minimal over all obstructions of $\phi$. For a polymorphism $\phi$, we denote by $O(\phi)$ a minimal obstruction of $\phi$ and by $r(\phi)$ the arity of the tuples in the minimal obstruction $O(\phi)$.

**Theorem 5.** Let $P$ be a nice polymorphism property such that all polymorphisms $\phi$ with $P(\phi)$ are idempotent. Then, STRONG $C_p$-BACKDOOR DETECTION is $W[2]$-hard parameterized by the size of the backdoor set, even for CSP instances over the Boolean domain.

Due to space constraints we omit the proof of the theorem. Since all min, max, majority, affine and Mal’cev polymorphisms can be defined via nice properties and are idempotent, we obtain the following corollary.

**Corollary 2.** For every $C \in \{\text{MIN, MAX, MAJ, AFF, MAL}\}$, STRONG $C$-BACKDOOR DETECTION is $W[2]$-hard parameterized by the size of the backdoor set, even for CSP instances over the Boolean domain.

In the following we show that hardness also holds if we drop the restriction on the domain of the CSP instance but instead consider only CSP instances of arity 2.

**Theorem 6.** For every $C \in \{\text{MIN, MAX, MAJ, AFF, MAL}\}$, STRONG $C$-BACKDOOR DETECTION is $W[2]$-hard even for CSP instances with arity 2.

The proof of the above theorem is inspired by the proof of (Bessiere et al. 2013) Theorem 6) and is omitted due to space constraints.

**Summary**

We have introduced heterogeneous base classes and have shown that strong backdoor sets into such classes can be considerably smaller than strong backdoor sets into homogeneous base classes; nevertheless, the detection of strong backdoor sets into homogeneous base classes is still fixed-parameter tractable in many natural cases. Hence our results push the tractability boundary considerably further. Our theoretical evaluation entails hardness results that establish the limits for tractability.

So far we have focused on strong backdoor sets; we leave a rigorous study of weak backdoor sets into heterogeneous base classes for future investigations. It would be interesting to extend our line of research to the study of backdoor sets into heterogeneous base classes composed of homogeneous classes defined by global properties, in contrast to the Shaefer classes of SAT, and polymorphism-based classes for CSP.

**Acknowledgments** Gaspers acknowledges support by the Australian Research Council, grant DE120101761. Szeider acknowledges support by the European Research Council, grant reference 239962 (COMPLEX REASON) and the Austrian Science Funds (FWF), project I836-N23 (ALGORITHMS AND COMPLEXITY OF CONSTRAINT LANGUAGES). Ordyniak acknowledges support from the Employment of Newly Graduated Doctors of Science for Scientific Excellence (CZ.1.07/2.3.00/30.0009). Živný is supported by a Royal Society University Research Fellowship.
References


