Backdoor Sets for CSP*

Serge Gaspers[1], Sebastian Ordyniak[2], and Stefan Szeider[3]

1 **UNSW Australia (The University of New South Wales), Sydney, Australia; and Data61 (formerly: NICTA), CSIRO, Australia**
sergeg@cse.unsw.edu.au

2 **Algorithms and Complexity Group, TU Wien, Vienna, Austria**
ordyniak@ac.tuwien.ac.at

3 **Algorithms and Complexity Group, TU Wien, Vienna, Austria**
szeider@ac.tuwien.ac.at

Abstract

A backdoor set of a CSP instance is a set of variables whose instantiation moves the instance into a fixed class of tractable instances (an island of tractability). An interesting algorithmic task is to find a small backdoor set efficiently: once it is found we can solve the instance by solving a number of tractable instances. Parameterized complexity provides an adequate framework for studying and solving this algorithmic task, where the size of the backdoor set provides a natural parameter. In this survey we present some recent parameterized complexity results on CSP backdoor sets, focusing on backdoor sets into islands of tractability that are defined in terms of constraint languages.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Backdoor sets, Constraint satisfaction problems, Parameterized complexity, Polymorphisms

Digital Object Identifier 10.4230/DFU.Vol7.15301.137

1 Introduction

The Constraint Satisfaction Problem (CSP) is a central and generic computational problem which provides a common framework for many theoretical and practical applications [32]. An instance of CSP consists of a collection of variables that must be assigned values subject to constraints, where each constraint is given in terms of a relation whose tuples specify the allowed combinations of values for specified variables. The problem was originally formulated by Montanari [47], and has been found equivalent to the homomorphism problem for relational structures [19] and the problem of evaluating conjunctive queries on databases [37]. In general, CSP is NP-complete. A central line of research is concerned with the identification of classes of instances for which CSP can be solved in polynomial time. Such classes are often called "islands of tractability" [37, 38].

A prominent way of defining islands of tractability for CSP is to restrict the relations that may occur in the constraints to a fixed set Γ, called a *constraint language*. A finite constraint language is *tractable* if CSP restricted to instances using only relations from Γ, denoted

* NICTA was funded by the Australian Government through the Department of Communications and the Australian Research Council (ARC) through the ICT Centre of Excellence Program. Serge Gaspers is the recipient of an ARC Future Fellowship (FT140100048) and acknowledges support under the ARC's Discovery Projects funding scheme (DP150101134). Stefan Szeider acknowledges the support by the Austrian Science Fund (FWF), grant reference P26696.

The Constraint Satisfaction Problem: Complexity and Approximability. *Dagstuhl Follow-Ups*, Volume 7, ISBN 978-3-95977-003-3.
Editors: Andrei Krokhin and Stanislav Živný; pp. 137–157
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

CSP(Γ), can be solved in polynomial time. Schaefer's famous Dichotomy Theorem [52] identifies all islands of tractability in terms of tractable constraint languages over the two-element domain. Since then, many extensions and generalizations of this result have been obtained [11, 34, 39, 53]. The Dichotomy Conjecture of Feder and Vardi [18] states that for every finite constraint language Γ, CSP(Γ) is either NP-complete or solvable in polynomial time. Schaefer's Dichotomy Theorem shows that the conjecture holds for two-element domains; more recently, Bulatov [4] showed the conjecture to be true for three-element domains.

If a CSP instance does not belong to a known island of tractability, none of the above tractability results apply. What if an instance does not belong to an island, but is "close" to an island in a certain way? Can we exploit this closeness algorithmically and possibly scale the island's tractability to the considered instance? In order to answer this question one needs to provide a definition for the distance of a CSP instance from an island of tractability (or, more generally, from a class $\mathcal{H}$ of CSP instances). The notion of (strong or weak) backdoor sets, introduced by Williams et al. [55], provides natural distance measures. A *strong backdoor set* of a CSP instance $\mathcal{I}$ into a class $\mathcal{H}$ of CSP instances is a set B of variables of $\mathcal{I}$ such that for all instantiations of the variables in B, the reduced instance belongs to $\mathcal{H}$ (we provide a more formal definition in Section 2.5). The set B is a *weak backdoor set* if for at least one instantiation the reduced instance is satisfiable and belongs to $\mathcal{H}$.

Once we know a strong backdoor set of size k into an island of tractability $\mathcal{H}$ for a CSP instance $\mathcal{I}$ over a finite domain of size d, we can solve $\mathcal{I}$ by solving at most d^k many tractable instances that arise by all the possible instantiations of the backdoor set ($\mathcal{I}$ is satisfiable if and only if at least one of the reduced instances is). Similarly, if we know a weak backdoor set of size k, we can actually find a satisfying assignment again by solving all reduced instances that arise from instantiating the backdoor set and belong to $\mathcal{H}$. The size of a smallest backdoor set provides a notion of distance of the instance from $\mathcal{H}$.

Overall, if we can solve an instance $\mathcal{I}$ of a class $\mathcal{H}$ in polynomial time, say in time $p(|\mathcal{I}|)$, then we can solve an instance for which we know a strong backdoor set of size k into $\mathcal{H}$ in time $d^k p(|\mathcal{I}|)$. This is an exponential running time with the special feature that it is exponential not in the instance size $|\mathcal{I}|$, but in the domain size and backdoor set size only. Problems that admit solution of this type are called *fixed-parameter tractable* [13]. In fact, fixed-parameter tractability provides a desirable way of scaling with the parameter (in this case, the backdoor set size), much preferred over a scaling of the form $|\mathcal{I}|^k$, where the order of the polynomial depends on k.

The *backdoor set approach* for CSP consistes of two parts, first finding a possibly small backdoor set and second using the backdoor set to solve the CSP instance.

This brings up the question: under which circumstances one can efficiently detect a weak or strong backdoor set of size at most k, if one exists? Stated more specifically: under which circumstances is the problem of detecting a weak or strong backdoor set fixed-parameter tractable when parameterized by the size of a smallest backdoor set?

The systematic study of the parameterized complexity of this backdoor set detection problem in the context of propositional satisfiability (SAT) was initiated by Nishimura et al. [48], and has since then received a lot of attention (see the survey paper [29]). Over the last few years, this research has been extended to the area of CSP and the present article provides a survey for some of the results in this direction. Namely we will focus on strong backdoor sets into classes of CSP instances defined via restrictions on the allowed constraint languages.

This survey is structured as follows: In Section 2 we provide the preliminaries about CSP, parameterized complexity, and the backdoor set approach. We also show the first very general results about the application of the backdoor sets approach to CSP that will provide the skeleton for the results in the remaining sections. Sections 3–6 cover the main results of this survey, i.e., they cover the application of strong backdoor sets to CSP using more and more general and evolved base classes defined via restrictions on the constraint language. Starting with base classes defined via a single constraint language in Section 3 the exposition goes on to cover bases classes defined via finite and infinite sets of constraint languages in Sections 4 and 5. The results presented in Sections 3 and 4 are based on [28] and the results from Section 5 are based on [26]. Section 5.1 then gives an overview how the results in Section 5 can be applied for Valued CSP and are based on [25]. Section 6 is based on recent work [27] and outlines how even large backdoor sets can be exploited, as long as the backdoor set induces a graph with a sufficiently simple structure. Section 7 is devoted to a brief exposition of related work. We conclude in Section 8.

2 Preliminaries

2.1 Constraint Satisfaction

Let $\mathcal{V}$ be an infinite set of variables and $\mathcal{D}$ a finite set of values. A *constraint of arity ρ over $\mathcal{D}$* is a pair (S, R) where $S = (x_1, \dots, x_\rho)$ is a sequence of variables from $\mathcal{V}$ and $R \subseteq \mathcal{D}^\rho$ is a ρ-ary relation. The set $\text{var}(C) = \{x_1, \dots, x_\rho\}$ is called the *scope* of C. A *value assignment* (or *assignment*, for short) $\alpha : X \to \mathcal{D}$ is a mapping defined on a set $X \subseteq \mathcal{V}$ of variables. An assignment $\alpha : X \to \mathcal{D}$ *satisfies* a constraint $C = ((x_1, \dots, x_\rho), R)$ if $\text{var}(C) \subseteq X$ and $(\alpha(x_1), \dots, \alpha(x_\rho)) \in R$. For a set of constraints $\mathcal{I}$ we write $\text{var}(\mathcal{I}) = \bigcup_{C \in \mathcal{I}} \text{var}(C)$ and $\text{rel}(\mathcal{I}) = \{ R : (S, R) \in C, C \in \mathcal{I} \}$.

A finite set $\mathcal{I}$ of constraints is *satisfiable* if there exists an assignment that satisfies all the constraints in $\mathcal{I}$. The *Constraint Satisfaction Problem* (CSP, for short) asks, given a finite set $\mathcal{I}$ of constraints, whether $\mathcal{I}$ is satisfiable. Therefore we refer to a finite set of constraints also as a *CSP instance*.

Let $\alpha : X \to \mathcal{D}$ be an assignment. For a ρ-ary constraint $C = (S, R)$ with $S = (x_1, \dots, x_\rho)$ we denote by $C|_\alpha$ the constraint (S', R') obtained from C as follows. R' is obtained from R by (i) deleting all tuples $(d_1, \dots, d_\rho)$ from R for which there is some $1 \le i \le \rho$ such that $x_i \in X$ and $\alpha(x_i) \neq d_i$, and (ii) removing from all remaining tuples all coordinates d_i with $x_i \in X$. S' is obtained from S by deleting all variables x_i with $x_i \in X$. For a set $\mathcal{I}$ of constraints we define $\mathcal{I}|_\alpha$ as $\{ C|_\alpha : C \in \mathcal{I} \}$.

A *constraint language* (or *language*, for short) Γ over a finite domain $\mathcal{D}$ is a set Γ of relations (of possibly various arities) over $\mathcal{D}$. By CSP(Γ) we denote CSP restricted to instances $\mathcal{I}$ with $\text{rel}(\mathcal{I}) \subseteq \Gamma$. A constraint language Γ is *globally tractable* if there is a polynomial-time algorithm solving any CSP instance $\mathcal{I} \in \text{CSP}(\Gamma)$ in polynomial time. A constraint language Γ is *efficiently recognizable* if there is an algorithm, which, for any CSP instance $\mathcal{I}$, determines whether $\text{rel}(\mathcal{I}) \subseteq \Gamma$ in time polynomial in $|\mathcal{I}|$. Clearly, every finite constraint language is efficiently recognizable.

2.2 Polymorphisms

Here we introduce the concept of polymorphism, one of the most common ways to define infinite (tractable) constraint languages. Let $\mathcal{D}$ be a finite set of values, ρ and n be natural numbers, let $R \subseteq \mathcal{D}^\rho$, and let $t \in R$. We denote by $t[i]$ the i-th coordinate of t, where i is a natural number with $1 \le i \le \rho$. A *n-ary operation over $\mathcal{D}$* is a function from $\mathcal{D}^n$ to $\mathcal{D}$. We say R is *closed* under some n-ary operation φ over $\mathcal{D}$ if R contains the tuple

$\langle\varphi(t_1[1],\ldots,t_n[1]),\ldots,\varphi(t_1[\rho],\ldots,t_n[\rho])\rangle$ for every sequence $t_1,\ldots,t_n$ (of not necessarily distinct) tuples in R.

For a sequence $t_1,\ldots,t_n$ of tuples of a relation R we will often write $\varphi[t_1,\ldots,t_n]$ to denote the tuple $\langle\varphi(t_1[1],\ldots,t_n[1]),\ldots,\varphi(t_1[\rho],\ldots,t_n[\rho])\rangle$. The operation φ is also said to be a *polymorphism of* R.

Let φ be an n-ary operation over $\mathcal{D}$. We denote by $\Gamma(\varphi)$ the constraint language over $\mathcal{D}$ consisting of all relations that are closed under φ and we write $\mathrm{CSP}(\varphi)$ as an abbreviation for $\mathrm{CSP}(\Gamma(\varphi))$. We say that the operation φ is *tractable* if $\Gamma(\varphi)$ is globally tractable. We extend the notion of closedness under the polymorphism φ from relations to constraints and entire CSP instances in the natural way, i.e., we say that a constraint $C = (S, R)$ of a CSP instance $\mathcal{I}$ is closed under the operation φ if $R \in \Gamma(\varphi)$ and we say that the same applies to the CSP instance $\mathcal{I}$ if $\mathcal{I} \in \mathrm{CSP}(\varphi)$.

2.3 Base Classes

As base classes for the backdoor set approach we use classes of CSP instances that are defined via (possibly singular) sets of constraint languages. We will consider two basic types of constraint languages, finite and infinite constraint languages. Whereas finite constraint languages will always be represented explicitly, we will characterize infinite constraint languages via polymorhisms and sets of infinite constraint languages via types of polymorphisms.

The following are well-known types of operations.

- An operation $\varphi : D \to D$ is *constant* if there is a $d \in D$ such that for every $d' \in D$, it holds that $\varphi(d') = d$;
- An operation $\varphi : D^n \to D$ is *idempotent* if for every $d \in D$ it holds that $\varphi(d,\ldots,d) = d$;
- An operation $\varphi : D^n \to D$ is *conservative* if for every $d_1,\ldots,d_n \in D$ it holds that $\varphi(d_1,\ldots,d_n) \in \{d_1,\ldots,d_n\}$;
- An operation $\varphi : D^2 \to D$ is a *min/max* operation if there is an ordering of the elements of D such that for every $d, d' \in D$, it holds that $\varphi(d,d') = \varphi(d',d) = \min\{d,d'\}$ or $\varphi(d,d') = \varphi(d',d) = \max\{d,d'\}$, respectively;
- An operation $\varphi : D^3 \to D$ is a *majority* operation if for every $d, d' \in D$ it holds that $\varphi(d,d,d') = \varphi(d,d',d) = \varphi(d',d,d) = d$;
- An operation $\varphi : D^3 \to D$ is an *minority* operation if for every $d, d' \in D$ it holds that $\varphi(d,d,d') = \varphi(d,d',d) = \varphi(d',d,d) = d'$;
- An operation $\varphi : D^3 \to D$ is a *Mal'cev* operation if for every $d, d' \in D$ it holds that $\varphi(d,d,d') = \varphi(d',d,d) = d'$.

We denote by VAL, MINMAX, MAJ, AFF, and MAL the classes of CSP instances $\mathcal{I}$, which are closed under some constant, min/max, majority, minority, or Mal'cev operation, respectively. These are some of the most well-known classes of tractable CSP instances and are closely related to (generalizations of) the well-known Schaefer languages [52].

When applying the backdoor set approach to base classes defined via sets of infinite constraint languages, it will become convenient to define more general types of operations than the standard ones introduced above. Namely, we will define predicates of operations (called tractable polymorphism predicates) allowing us to employ the backdoor set approach.

Let $\mathcal{P}(\varphi)$ be a predicate for the operation φ. We call $\mathcal{P}(\varphi)$ a *tractable polymorphism predicate* if the following conditions hold.

N1. There is a constant $c(\mathcal{P})$ such that for all finite domains D, all operations φ over D for which $\mathcal{P}$ holds are of arity at most $c(\mathcal{P})$.

N2. Given a operation φ and a domain D, one can check in polynomial time whether $\mathcal{P}(\varphi)$ holds on all of the at most $|D|^{c(\mathcal{P})}$ tuples over D,
N3. Every operation for which $\mathcal{P}$ holds is tractable.

For a tractable polymorphism predicate $\mathcal{P}$ we define $\Delta(\mathcal{P})$ to be the set of all constraint languages that are closed under some operation for which $\mathcal{P}$ holds. Note that the classes VAL, MINMAX, MAJ, AFF, and MAL as well as combination of these classes can easily be defined via tractable polymorphism predicates. Moreover also much more general types of operations such as semilattice operations (sometimes called ACI operation) [34] (a generalization of min/max), k-ary near unanimity operations [35, 19] (a generalization of majority), k-ary edge operations [33] (a generalization of Mal'cev), and the two operations of arities three and four [40] that capture the bounded width property [1] (a generalization of semilattice and near unanimity operations) can be defined via tractable polymorphism predicates. Finally, we would like to note here that the property of belonging to a tractable algebraic variety [5] is an example of a tractable polymorphism predicate.

2.4 Parameterized Complexity

A *parameterized problem* Π is a problem whose instances are tuples $(\mathcal{I}, k)$, where $k \in \mathbb{N}$ is called the parameter. We say that a parameterized problem is *fixed parameter tractable* (FPT in short) if it can be solved by an algorithm which runs in time $f(k) \cdot |\mathcal{I}|^{\mathcal{O}(1)}$ for some computable function f; algorithms with a running time of this form are called FPT algorithms. FPT also denotes the class of all fixed-parameter tractable decision problems. The notions of W[i]*-hardness* (for $i \in \mathbb{N}$) are frequently used to show that a parameterized problem is not likely to be FPT. The W[i] classes and FPT are closed under parameterized reductions, which are FPT algorithms reducing any instance $(\mathcal{I}, k)$ of a parameterized problem Π to an instance $(\mathcal{I}', k')$ of a parameterized problem Π' such that $(\mathcal{I}, k)$ is a yes-instance for Π if and only if $(\mathcal{I}', k')$ is a yes-instance for Π', and k' is upper bounded by a function of k. An FPT algorithm for a W[i]-hard problem would imply that the Exponential Time Hypothesis fails [9]. We refer the reader to other sources [14, 23] for an in-depth introduction to parameterized complexity.

We also consider parameterized problems with multiple parameters $k_1, \dots, k_\ell$ or parameterized by a set $T = \{k_1, \dots, k_\ell\}$ of natural numbers, whose instances are tuples $(\mathcal{I}, T)$, where $k_i \in \mathbb{N}$, $1 \leq i \leq \ell$, are the parameters. Such parameterized problems are equivalent to parameterized problems with a single parameter $k_1 + \cdots + k_\ell$.

2.5 Backdoors

Let $\mathcal{I}$ be an instance of CSP over $\mathcal{D}$ and let $\mathcal{H}$ be a class of CSP instances. A set B of variables of $\mathcal{I}$ is called a *strong backdoor set* into $\mathcal{H}$ (or shortly *strong $\mathcal{H}$-backdoor set*) if for every assignment $\alpha : B \to \mathcal{D}$ it holds that $\mathcal{I}|_\alpha \in \mathcal{H}$. Notice that if we are given a strong backdoor set B of size k into a class $\mathcal{H}$ of CSP instances that can be solved in polynomial time, then it is possible to solve the entire instance in time $|\mathcal{D}|^k \cdot |\mathcal{I}|^{\mathcal{O}(1)}$. In general for a class $\mathcal{H}$ of CSP instances the application of the backdoor set approach usually requires the solution to the following two subproblems.

Strong $\mathcal{H}$-Backdoor Set Detection
Input: A CSP instance $\mathcal{I}$ over the same domain as $\mathcal{H}$ and a non-negative integer k.
Question: Find a strong $\mathcal{H}$-backdoor set for $\mathcal{I}$ of cardinality at most k, or determine that no such strong backdoor set exists.

STRONG $\mathcal{H}$-BACKDOOR SET EVALUATION **Input:** A CSP instance $\mathcal{I}$ over the same domain as $\mathcal{H}$ and a strong $\mathcal{H}$-backdoor set for $\mathcal{I}$. **Question:** Determine whether $\mathcal{I}$ is satisfiable.

We will consider the parameterized complexity of the above problems depending on various base classes $\mathcal{H}$ defined via restrictions on the allowed constraint languages as well as the following parameters:

- `arity` denoting the maximum arity of the given CSP instance,
- `dom` denoting the maximum domain of the given CSP instance, and
- `bd-size` denoting the bound on the backdoor set size given as an input to the STRONG $\mathcal{H}$-BACKDOOR SET DETECTION problem or the size of the backdoor set given as an input to the STRONG $\mathcal{H}$-BACKDOOR SET EVALUATION problem, respectively.
- $\texttt{bd-size}_{\mathcal{H}}$ denoting the smallest size of a strong $\mathcal{H}$-backdoor set for the provided CSP instance.

As it turns out for all the results surveyed here, the complexity of the above problems solely depends on the following two properties of the base class $\mathcal{H}$. For a set $T \subseteq \{\texttt{arity}, \texttt{dom}, \texttt{bd-size}\}$, we say that a class of CSP instances $\mathcal{H}$ is:

- *T-tractable* if there is an FPT-algorithm parameterized by T that solves every CSP instance in $\mathcal{H}$.
- *T-detectable* if there is an FPT algorithm parameterized by T, denoted by $\mathcal{A}_{\mathcal{H}}$, that, given a CSP instance $\mathcal{I}$ and a set $B \subseteq \text{var}(\mathcal{I})$, determines whether B is a strong $\mathcal{H}$-backdoor set of $\mathcal{I}$, and if not, outputs a set $Q \subseteq \text{var}(\mathcal{I}) \setminus B$ whose size can be bounded by a function of T, such that every strong $\mathcal{H}$-backdoor set of $\mathcal{I}$ containing B contains at least one variable in Q.

▶ **Theorem 1.** STRONG $\mathcal{H}$-BACKDOOR SET EVALUATION *is fixed-parameter tractable parameterized by $T \cup \{\texttt{dom}, \texttt{bd-size}\}$ for every T-tractable class $\mathcal{H}$ of CSP instances.*

Proof. We solve such an instance $\mathcal{I} \in \mathcal{H}$ by going over all of the at most $\texttt{dom}^{\texttt{bd-size}}$ assignments of the variables in the given backdoor set and checking for each of those whether the reduced instance is satisfiable by using the algorithm implied because $\mathcal{H}$ is T-tractable. ◀

▶ **Theorem 2.** STRONG $\mathcal{H}$-BACKDOOR SET DETECTION *is fixed-parameter tractable parameterized by $T \cup \{\texttt{bd-size}\}$ for every T-detectable class $\mathcal{H}$ of CSP instances.*

Proof. We will employ a branching algorithm that employs $\mathcal{A}_{\mathcal{H}}$ as a subroutine. The main ingredient of the algorithm is a recursive function, which is called with a set B of at most k variables representing a partial backdoor set. The algorithm simply returns the value of the recursive function called with the empty set of variables and the recursive function consists of the following steps.

1. The function executes the algorithm $\mathcal{A}_{\mathcal{H}}$ on $\mathcal{I}$ and B.
2. If Step 1 concludes that B is a strong $\mathcal{H}$-backdoor set, then the function returns YES,
3. otherwise, let Q be the set of variables returned by the algorithm $\mathcal{A}_{\mathcal{H}}$ in Step (1). Then,
   - if $|B| = k$, the function returns NO,
   - otherwise the function branches on all variables in Q, i.e., for every variable $q \in Q$, the function calls itself on the set $B \cup \{q\}$. If any of these calls returns YES, then the function returns YES, otherwise it returns NO.

This concludes the description of the algorithm. Its correctness follows from the properties of algorithm $\mathcal{A}_{\mathcal{H}}$. Because of the properties of the algorithm $\mathcal{A}_{\mathcal{H}}$ the number of times that the recursive function calls itself recursively is bounded by a function of T. Moreover since the recursive function does not call itself when $|B| = k$, the depth of the recursion is at most k. It follows that the total number of calls to the recursive function is bounded by a function of $T \cup \{k\}$. Finally, the time required for each call of the recursive function is dominated by the time required by $\mathcal{A}_{\mathcal{H}}$. This shows that SBD($\mathcal{H}$) is fixed-parameter tractable parameterized by $T \cup \{\texttt{bd-size}\}$. ◀

3 Basic Results

In this section we consider the backdoor set approach for classes $\mathcal{H}$ of CSP instances defined by a single constraint language Γ. It is easy to see that any such class $\mathcal{H}$ is $\emptyset$-tractable if and only if the defining constraint language Γ is globally tractable. Hence we obtain from Theorem 1 that if Γ is globally tractable, then Strong $\mathcal{H}$-Backdoor Set Evaluation is fixed-parameter tractable parameterized by $\{\texttt{dom}, \texttt{bd-size}\}$. We will now show that if Γ is additionally efficiently recognizable, then Strong $\mathcal{H}$-Backdoor Set Detection is fixed-parameter tractable parameterized by $\{\texttt{arity}, \texttt{bd-size}\}$. Because of Theorem 2 it is sufficient to show that $\mathcal{H}$ is $\{\texttt{arity}\}$-detectable.

▶ **Lemma 3.** CSP(Γ) *is* $\{\texttt{arity}\}$*-detectable for every efficiently recognizable constraint language* Γ*.*

Proof. Let $\mathcal{H} = \text{CSP}(\Gamma)$. It is sufficient to give the algorithm $\mathcal{A}_{\mathcal{H}}$. Let $\mathcal{I}$ be the given CSP instance, B be the given set of variables of $\mathcal{I}$, let m and t be the number of constraints and the maximum number of tuples in any constraint of $\mathcal{I}$, respectively. Observe that B is a strong $\mathcal{H}$-backdoor set if and only if for every constraint $C = (S, R)$ of $\mathcal{I}$ it holds that $C|_\alpha \in \mathcal{H}$ for every assignment α of the variables in $B \cap S$. By ordering the tuples in R according to the assignments of the variables in $B \cap S$, this can be checked in time $O(t \log t)$ times the time required to determine whether $\mathcal{I} \in \mathcal{H}$. Hence executing this for every constraint requires $O(m \cdot t \log t \cdot |\mathcal{I}|^{O(1)})$ time. If $C|_\alpha \in \mathcal{H}$ for every constraint $C = (S, R)$ and every such assignment of the variables in $B \cap S$, then the algorithm returns Yes. Otherwise there is a constraint $C = (S, R)$ and an assignment α of the variables in $B \cap S$ such that $C|_\alpha \notin \mathcal{H}$. In this case B is not a strong $\mathcal{H}$-backdoor set and the algorithm returns the set $S \setminus B$, which we claim satisfies the properties of the set Q given in the statement of the algorithm. Towards showing this, assume for a contradiction that this is not the case, i.e., there is a strong $\mathcal{H}$-backdoor set B' with $B \subseteq B'$ that does not contain a variable in $S \setminus B$. Note that because $B \subseteq B'$ and B' does not contain any variable in $S \setminus B$, it holds that $B' \cap S = B \cap S$. Hence the assignment α as given above contradicts our assumption that B' is a strong $\mathcal{H}$-backdoor set, because $C|_\alpha \notin \mathcal{H}$. ◀

As an immediate consequence of the above lemma and Theorem 2 we obtain the following.

▶ **Theorem 4.** Strong CSP(Γ)-Backdoor Set Detection *is fixed-parameter tractable parameterized by* $\{\texttt{arity}, \texttt{bd-size}\}$ *for every efficiently recognizable constraint language* Γ*.*

This leads to our main result for base classes $\mathcal{H}$ defined via single constraint languages.

▶ **Corollary 5.** *CSP is fixed-parameter tractable parameterized by* $\{\texttt{arity}, \texttt{dom}, \texttt{bd-size}_{\mathcal{H}}\}$ *for every efficiently recognizable and globally tractable constraint language* Γ*.*

If the constraint language Γ is finite, then the above result can even be improved to fixed-parameter tractability with respect to the single parameter backdoor set size. This is because any finite constraint language has bounded domain and bounded arity, i.e., bounded by some fixed constants say D and R, respectively. Hence any input CSP instance with domain larger than D or with arity at least $R + k$ (where k is the size of the backdoor set) can immediately be identified as a No-instance and thus the above algorithm only needs to be applied to CSP instances of domain at most D and arity at most $R + k$. Since all finite constraint languages are efficiently recognizable, we obtain the following corollary.

▶ **Corollary 6.** *CSP is fixed-parameter tractable parameterized by* $\texttt{bd-size}_{\text{CSP}(\Gamma)}$ *for every globally tractable finite constraint language* Γ*.*

In the following we show that when considering infinite constraint languages (in particular constraint languages defined via polymorphisms), then it is not possible to drop the arity parameter. In particular, we will show that Strong CSP(φ)-Backdoor Set Detection is not fixed-parameter tractable parameterized by the size of the backdoor set alone for any tractable idempotent operation φ.

▶ **Theorem 7.** Strong CSP(φ)-Backdoor Set Detection *is fixed-parameter intractable (*W[2]*-hard) parameterized by* `bd-size` *for every tractable idempotent operation* φ*, even for CSP instances over the Boolean domain.*

Proof. We start by introducing what we call "Boolean barriers" of operations since they form the basis of the proof. Let $\varphi : D^n \to D$ be an n-ary operation over D. We say a set λ of $r(\lambda)$-ary tuples over $\{0, 1\}$ is a *Boolean barrier* for φ if there is a sequence $\langle t_1, \ldots, t_n\rangle$ of (not necessarily distinct) tuples in λ such that $\varphi(t_1, \ldots, t_n) \notin \lambda$. We call a Boolean barrier λ of φ *minimal* if $|\lambda|$ is minimal over all Boolean barriers of φ. For an operation φ, we denote by $\lambda(\varphi)$ a minimal Boolean barrier of φ. For our reduction below, we will employ the fact that every tractable operation φ has a non-empty Boolean barrier of finite size. The reason that a Boolean barrier must exist is simply because if φ would not have a Boolean barrier then every Boolean CSP instance would be closed under φ and thus tractable, which unless $\mathrm{P} = \mathrm{NP}$ is not possible. To see that $\lambda(\varphi)$ is finite first note that $|\lambda(\varphi)|$ is at most as large as the arity r_φ of φ. Moreover, $r(\lambda) \le 2^{r_\varphi}$ because there are at most 2^{r_φ} distinct r_φ-ary tuples over $\{0, 1\}$. Hence the size of $\lambda(\varphi)$ is at most $r_\varphi \cdot 2^{r_\varphi}$.

We are now ready to show the theorem via a parameterized reduction from the well-known W[2]-hard Hitting Set problem [13]. Let $\langle U, \mathcal{F}, k\rangle$ be an instance of Hitting Set, where U is a set (often refered to as the universe), $\mathcal{F}$ is a familly of subsets of U and k is a non-negative integer. Note that the Hitting Set problem asks whether there is a subsets $H \subseteq U$ of the universe U with cardinality and most k such that $H \cap F \neq \emptyset$ for every $F \in \mathcal{F}$. We construct a CSP instance $\mathcal{I}$ such that $(U, \mathcal{F})$ has a hitting set of size at most k if and only if $\mathcal{I}$ has a strong CSP(φ)-backdoor set of size at most k.

In the following let $\lambda(\varphi) = \{t_1, \ldots, t_n\}$ and r denote the arity of the tuples in $\lambda(\varphi)$. The variables of $\mathcal{I}$ are $\{\, x_u : u \in U \,\} \cup \{\, o_1(F), \ldots, o_r(F) : F \in \mathcal{F} \,\}$. Furthermore, for every $F \in \mathcal{F}$ with $F = \{u_1, \ldots, u_{|F|}\}$, C contains a constraint $R(F)$ with scope $\langle o_1(F), \ldots, o_r(F), x_{u_1}, \ldots, x_{u_{|F|}}\rangle$ whose relation contains the row

$$t_i[1], \ldots, t_i[r], \underbrace{\langle i \bmod 2, \ldots, i \bmod 2\rangle}_{|F| \text{ times}}$$

for every i in $1 \le i \le n$. This completes the construction of $\mathcal{I}$. Suppose that $\mathcal{F}$ has a hitting set B of size at most k. We claim that $B_u = \{\, x_u : u \in B \,\}$ is a strong CSP(φ)-backdoor set of $\mathcal{I}$. Let α be an assignment of the variables in B. We claim that $\mathcal{I}|_\alpha$ is closed under φ and hence B_u is a strong CSP(φ)-backdoor set of $\mathcal{I}$. Note that because φ is idempotent every relation containing only a single tuple is closed under φ, it holds that $|\lambda(\varphi)| > 1$. Because B

is a hitting set of H, it follows that every relation of $\mathcal{I}|_\alpha$ contains at least one tuple less than the corresponding relation in $\mathcal{I}$. Hence any relation of $\mathcal{I}|_\alpha$ contains less tuples than $|\lambda(\varphi)|$, which is the minimal size of any boolean barrier for φ, which implies that $\mathcal{I}|_\alpha$ is closed under φ, as required.

For the reverse direction, suppose that $\mathcal{I}$ has a strong CSP(φ)-backdoor set B of size at most k. Because no constraint of $\mathcal{I}$ is closed under φ, we obtain that B has to contain at least one variable from every constraint of $\mathcal{I}$. Since the only variables that are shared between $R(F)$ and $R(F')$ for distinct $F, F' \in \mathcal{F}$ are the variables in $\{ x_u : u \in U \}$, it follows that B is a hitting set of size at most k for $\mathcal{F}$, as required. ◀

4 Heterogeneous Base Classes

In the previous section, we considered so-called *homogeneous* base classes defined via a single globally tractable constraint language. In this section we introduce a more general form of base classes (called *heterogeneous* base classes) that are defined via a set of globally tractable constraint languages. In particular, given a set of globally tractable constraint languages Δ, let CSP(Δ) be the class of all CSP instances in $\bigcup_{\Gamma\in\Delta}$ CSP(Γ). The size of a backdoor set into a heterogeneous base class can be much smaller than the minimum size of a backdoor set into any of its homogeneous base classes. Therefore, backdoor sets into heterogeneous base classes are considerably more powerful but also more complicated to handle. Even the evaluation of backdoor sets into heterogeneous base classes needs to be handled carefully. Namely, because the given set Δ can be infinite the class CSP(Δ) of CSP instances is not a priory $\emptyset$-tractable even if the considered constraint languages are globally tractable.

We start by showing that backdoor sets into heterogeneous base classes can be arbitrarily smaller than backdoor sets into their homogeneous counterparts. For the construction of the example let φ_{min} be the *min*-type operation and let φ_{maj} be the *majority*-type operation both defined on the ordered domain $(0, 1)$.

▶ **Proposition 8.** *For every natural number n, there is a CSP instance $\mathcal{I}_n$ such that $\mathcal{I}_n$ has a strong* CSP($\{\varphi_{\text{min}}, \varphi_{\text{maj}}\}$)*-backdoor set of size one but every strong* CSP($\{\varphi_{\text{min}}\}$)*-backdoor set and every strong* CSP($\{\varphi_{\text{maj}}\}$)*-backdoor set of $\mathcal{I}_n$ has size at least n.*

Proof. Let MAJ$[a, b, c, d]$ be the constraint with scope (a, b, c, d) and whose relation contains all possible tuples that set d to 0 except for the tuple $(1, 1, 1, 0)$. Then MAJ$[a, b, c, d]$ is not closed under φ_{maj} but happens to be closed under φ_{min}. Similarly, let MIN$[a, b, c]$ be the constraint with scope (a, b, c) and whose relation contains the tuples $(0, 1, 1)$ and $(1, 0, 1)$. Then MIN$[a, b, c]$ is not closed under φ_{min} but happens to be closed under φ_{maj}. We claim that the CSP instance $\mathcal{I}_n$ with variables $\{y_1, \dots, y_{3n}\} \cup \{z_1, \dots, z_{2n}\} \cup \{x\}$ and constraints $\text{MAJ}_i = \text{MAJ}[y_{3i+1}, y_{3i+2}, y_{3i+3}, x]$ and $\text{MIN}_i = \text{MIN}[z_{2i+1}, z_{2i+2}, x]$ for every i with $0 \leq i \leq n-1$, satisfies the claim of the proposition. Towards showing this first note that $\{x\}$ is a strong CSP($\{\varphi_{\text{min}}, \varphi_{\text{maj}}\}$)-backdoor set of $\mathcal{I}_n$ of size 1. This is because for the assignment α with $\alpha(x) = 0$, it holds that $\mathcal{I}_n|_\alpha \in$ CSP(φ_{min}) and for the assignment α with $\alpha(x) = 1$, it holds that $\mathcal{I}_n|_\alpha \in$ CSP(φ_{maj}). Moreover it is straightforward to verify that every strong CSP(φ_{min})-backdoor set of $\mathcal{I}_n$ has to contain at least one variable from every constraint MIN_i that is not x and similarly every strong CSP(φ_{maj})-backdoor set of $\mathcal{I}_n$ has to contain at least one variable from every constraint MAJ_i that is not x. Hence the size of every strong CSP(φ_{min})-backdoor set as well as any strong CSP(φ_{maj})-backdoor set is at least n. ◀

The following auxiliary lemma provides a useful property for detecting backdoor sets into heterogeneous base classes.

▶ **Lemma 9.** *Let Δ be a set of constraint languages, $\mathcal{I}$ a CSP instance, and B a set of variables of $\mathcal{I}$. If there is an assignment α of the variables in B such that $\mathcal{I}|_\alpha \notin \mathrm{CSP}(\Delta)$, then any strong* CSP($\Delta$)*-backdoor set B' with $B \subseteq B'$ contains at least one variable from the set $Q = (\bigcup_{\Gamma \in \Delta} \mathrm{var}(C^\Gamma)) \setminus B$, where for any $\Gamma \in \Delta$, C^Γ is a constraint of $\mathcal{I}|_\alpha$ such that $C^\Gamma \notin \mathrm{CSP}(\Gamma)$.*

Proof. Let $\mathcal{H} = \mathrm{CSP}(\Delta)$. Assume for a contradiction that this is not the case, i.e., there is a strong $\mathcal{H}$-backdoor set B' for $\mathcal{I}$ with $B \subseteq B'$ and B' does not contain any variable from Q. Because B is not a strong $\mathcal{H}$-backdoor set there is an assignment α such that $\mathcal{I}|_\alpha \notin \mathcal{H}$. Let α' be any assignment of the variables in B' that agrees with α on the variables in B. Because B' is a strong $\mathcal{H}$-backdoor set, it follows that there is a constraint language $\Gamma \in \Delta$ such that $\mathcal{I}|_{\alpha'} \in \mathrm{CSP}(\Gamma)$. We claim that B' contains at least one variable from every constraint $C \in \mathcal{I}|_\alpha$ with $C \notin \mathrm{CSP}(\Gamma)$. For if not, then let C be such a constraint with an empty intersection with B'. It follows that $C \in \mathcal{I}|_{\alpha'}$ but because $C \notin \mathrm{CSP}(\Gamma)$ this contradicts our assumption that $\mathcal{I}|_{\alpha'} \in \mathrm{CSP}(\Gamma)$. ◀

In the following we will give our first example of a heterogeneous base class, which can be employed for the backdoor set approach. Namely, we will show this for any finite set Δ of finite globally tractable constraint languages. We start by showing that CSP(Δ) is $\emptyset$-tractable, which, because of Theorem 1, implies that Strong CSP(Δ)-Backdoor Set Evaluation is fixed-parameter tractable parameterized by `bd-size` (because the domain is finite).

▶ **Proposition 10.** CSP(Δ) *is $\emptyset$-tractable for every finite set Δ of finite and globally tractable constraint languages.*

Proof. Let $\mathcal{H} = \mathrm{CSP}(\Delta)$. We can solve a CSP instance $\mathcal{I} \in \mathcal{H}$ by going over all constraint languages $\Gamma \in \Delta$, checking whether $\mathcal{I}$ is in CSP(Γ) and if so solving $\mathcal{I}$ in polynomial time. This can be achieved in polynomial time because there are only finitely many constraint languages in Δ and each of them can be recognized in polynomial time (because it is finite). ◀

The next lemma shows that for any such set Δ, CSP(Δ) is {`bd-size`}-detectable.

▶ **Lemma 11.** CSP(Δ) *is* {`bd-size`}*-detectable for every finite set Δ of finite constraint languages.*

Proof. It is sufficient to give the algorithm $\mathcal{A}_{\mathrm{CSP}(\Delta)}$. Let $\mathcal{I}$ be a CSP instance and let B be the given set of variables of $\mathcal{I}$. Because Δ contains only finitely many finite constraint languages, it holds that the maximum domain value D as well as the maximum arity R of any of its languages is also finite. Hence w.l.o.g. we can assume that the maximum domain value of $\mathcal{I}$ is at most D and similarly the maximum arity of $\mathcal{I}$ is at most $R + k$, since otherwise we can simply return No. To determine whether B is a strong CSP(Δ)-backdoor set, we test for every assignment α of the variables in B whether $\mathcal{I}|_\alpha \in \mathrm{CSP}(\Gamma)$ for some constraint language $\Gamma \in \Delta$. Because there are at most $D^{|B|}$ assignments of the variables in B, there are finitely many constraint languages in Δ, and verifying whether $\mathcal{I}|_\alpha \in \mathrm{CSP}(\Gamma)$ for some finite constraint language can be done in polynomial time, the total time required by this step of the algorithm is $O(D^{|B|}|\mathcal{I}|^{O(1)})$. If the above test holds for every assignment of the variables in B, then the algorithm returns Yes. Otherwise there is an assignment α of the variables in B such that $\mathcal{I}|_\alpha \notin \text{CSP}(\Delta)$. Hence we obtain from Lemma 9 that any strong CSP(Δ)-backdoor set B' with $B \subseteq B'$ contains at least one variable from the set Q as given in the statement of Lemma 9. Because the size of this set Q is at most $|\Delta|(R+k) \in O(k)$ and the set Q can be computed within the running time of the first step of the algorithm, the lemma follows. ◀

As an immediate consequence of the above lemma and Theorem 2, we obtain the following.

▶ **Theorem 12.** Strong CSP(Δ)-Backdoor Set Detection *is fixed-parameter tractable parameterized by* `bd-size` *for every finite set* Δ *of finite constraint languages.*

The above discussion leads directly to our main result for heterogeneous base classes defined via finite constraint languages.

▶ **Corollary 13.** *CSP is fixed-parameter tractable parameterized by* $\texttt{bd-size}_{CSP(\Delta)}$ *for every finite set* Δ *of globally tractable finite constraint languages.*

This concludes our discussion for heterogeneous base classes defined via finite constraint languages. In the following we will consider sets of infinite constraint languages defined via a tractable polymorphism predicate $\mathcal{P}$. Namely, let $\mathcal{H} = \text{CSP}(\Delta(\mathcal{P}))$ for a tractable polymorphism predicate $\mathcal{P}$. We will show that CSP is fixed-parameter tractable parameterized by `arity`, `dom`, and the size of a smallest strong $\mathcal{H}$-backdoor set. We will also show that none of these three parameters can be dropped without sacrificing fixed-parameter tractability. Crucial to this result is the fact that even though a tractable polymorphism predicate holds for a potentially infinite set of operations, the number of operations is bounded for a fixed domain.

▶ **Lemma 14.** *Let* $\mathcal{P}$ *be a tractable polymorphism predicate and* D *be a finite set. Then there are at most* $d^{d^{c(\mathcal{P})}}$ *operations on* D *that satisfy* $\mathcal{P}$*, and computing all these operations is fixed-parameter tractable parameterized by* $|D|$.

Proof. This follows because there are at most $d^{d^{c(\mathcal{P})}}$ $c(\mathcal{P})$-ary operations on D and because of Property N2 for each of those one can test in polynomial time whether it satisfies $\mathcal{P}$. ◀

We show next that Strong $\mathcal{H}$-Backdoor Set Evaluation is fixed-parameter tractable parameterized by $\{\texttt{dom}, \texttt{bd-size}\}$. Because of Theorem 1 it is sufficient to show that $\mathcal{H}$ is $\{\texttt{dom}\}$-tractable.

▶ **Lemma 15.** CSP($\Delta(\mathcal{P})$) *is* $\{\texttt{dom}\}$*-tractable for every tractable polymorphism predicate* $\mathcal{P}$.

Proof. We first compute the set P of all operations on the domain D of $\mathcal{I}$ using Lemma 14. We then check for every such operation $\varphi \in P$ whether $\mathcal{I}$ is closed under φ in time $O(m \cdot t^{c(\mathcal{P})})$, where m is the number of constraints of $\mathcal{I}$ and t is the maximum number of tuples occurring in any constraint of $\mathcal{I}$. If it is we use the fact that φ is tractable to solve $\mathcal{I}$ in polynomial time. ◀

We now turn to the detection of backdoor sets into $\mathcal{H}$.

▶ **Lemma 16.** CSP($\Delta(\mathcal{P})$) *is* $\{\texttt{arity}, \texttt{dom}, \texttt{bd-size}\}$*-detectable for every tractable polymorphism predicate* $\mathcal{P}$.

Proof. Let $\mathcal{H} = \text{CSP}(\Delta(\mathcal{P}))$. It is sufficient to give the algorithm $\mathcal{A}_\mathcal{H}$. Let $\mathcal{I}$ be the given CSP instance over $\mathcal{D}$ and let B be the given set of variables of $\mathcal{I}$.

We first compute the set P of all operations on $\mathcal{D}$ for which $\mathcal{P}$ holds by employing Lemma 14. Observe that a set B of variables of $\mathcal{I}$ is a strong $\mathcal{H}$-backdoor set if and only if it is a strong CSP(P)-backdoor set, where $\mathrm{CSP}(P) = \bigcup_{\varphi \in P} \mathrm{CSP}(\varphi)$.

To determine whether B is a strong CSP(P)-backdoor set, we test for every assignment α of the variables in B whether $\mathcal{I}|_\alpha \in \mathrm{CSP}(P)$. Because there are at most $\mathtt{dom}^{\mathtt{bd\text{-}size}}$ assignments of the variables in B, at most $\mathtt{dom}^{\mathtt{dom}^{c(\mathcal{P})}}$ operations in P, and verifying whether $\mathcal{I}|_\alpha \in \mathrm{CSP}(\varphi)$ for any $\varphi \in P$ can be achieved in polynomial time, the total time required by this step of the algorithm is at most $O(\mathtt{dom}^{\mathtt{bd\text{-}size}}\mathtt{dom}^{\mathtt{dom}^{c(\mathcal{P})}}|\mathcal{I}|^{O(1)})$. If the above holds for every assignment of the variables in B, then the algorithm returns YES. Otherwise there is an assignment α such that $\mathcal{I}|_\alpha \notin \mathrm{CSP}(P)$. Hence we obtain from Lemma 9 that any strong $\mathcal{H}$-backdoor set B' with $B \subseteq B'$ contains at least one variable from the set Q given in the statement of the lemma. Because the size of this set Q is at most $|P| \cdot \mathtt{arity} \leq \mathtt{dom}^{\mathtt{dom}^{c(\mathcal{P})}} \cdot \mathtt{arity}$ and the set Q can be computed within the running time of the first step of the algorithm, the lemma follows. ◀

The following theorem follows immediately from the above lemma and Theorem 2.

▶ **Theorem 17.** STRONG CSP($\Delta(\mathcal{P})$)-BACKDOOR SET DETECTION *is fixed-parameter tractable parameterized by* $\{\mathtt{arity}, \mathtt{dom}, \mathtt{bd\text{-}size}\}$ *for every tractable polymorphism predicate* $\mathcal{P}$.

The above results naturally lead to our main result of this section.

▶ **Corollary 18.** *CSP is fixed-parameter tractable parameterized by* $\{\mathtt{arity}, \mathtt{dom}, \mathtt{bd\text{-}size}_{\mathrm{CSP}(\Delta(\mathcal{P}))}\}$ *for every tractable polymorphism predicate* $\mathcal{P}$.

It turns out that we cannot avoid to parameterize by both the maximum domain value and the arity in the above theorem. We have already seen in Theorem 7 that even if we consider just a single idempotent tractable operation φ, then STRONG CSP(φ)-BACKDOOR SET DETECTION is fixed-parameter intractable parameterized by $\{\mathtt{bd\text{-}size}, \mathtt{dom}\}$ (even for boolean domain). Because it is easy to define a tractable polymorphism predicate that only holds for a single idempotent operation, this result generalizes to tractable polymorphism predicates. Hence it only remains to consider the case where we parameterize only by the maximum arity and backdoor set size. The proof of the following theorem can be found in [28, Theorem 12]. Note that since the reduction employed in the hardness result does strongly depend on the considered tractable polymorphism predicate it is difficult to obtain a general result as in Theorem 7, but it can be stated to include some of the arguably most prominent types of operations.

▶ **Theorem 19.** STRONG $\mathcal{H}$-BACKDOOR SET DETECTION *is fixed-parameter intractable (*W[2]*-hard) parameterized by* $\{\mathtt{arity}, \mathtt{bd\text{-}size}\}$ *for every* $\mathcal{H} \in$ {MINMAX*,* MAJ*,* AFF*,* MAL}*, even for CSP instances with arity* 2.

5 Scattered Base Classes

Thus far we have considered a setting in which the reduced CSP instance for each assignment of the backdoor set variables belonged entirely to a single tractable constraint language. To ensure that the reduced CSP instance is tractable, which is sufficient for an application of the backdoor set approach, there is however a natural and more general possibility: Instead of belonging entirely to a single tractable constraint language, the CSP instance could consist of a disjoint union of (pairwise variable disjoint) CSP instances, each belonging to some tractable constraint language. This type of tractable base class is particularly interesting in combination with the backdoor set approach, since now the variables of the backdoor set can be naturally employed to separate the CSP instance into parts only interacting with each other through the variables in the backdoor set.

More specifically, a CSP instance $\mathcal{I}$ is *connected* if either it consists of at most one constraint, or for each partition of $\mathcal{I}$ into nonempty sets $\mathcal{I}_1$ and $\mathcal{I}_2$, there exists at least one constraint $c_1 \in \mathcal{I}_1$ and one constraint $c_2 \in \mathcal{I}_2$ that share at least one variable. A *connected component* of $\mathcal{I}$ is a maximal connected subinstance $\mathcal{I}'$ of $\mathcal{I}$. These notions naturally correspond to the connectedness and connected components of standard graph representations of CSP instances.

Now, given a set Δ of constraint languages, we define the scattered class $\oplus(\Delta)$ of CSP instances $\mathcal{I}$ where each connected component $\mathcal{I}'$ of $\mathcal{I}$ belongs to $\mathrm{CSP}(\Gamma)$ for some $\Gamma \in \Delta$.

We start by showing that backdoor sets into scattered base classes can be arbitrarily smaller than backdoor sets into their heterogeneous counterparts. For the construction of the example let $\varphi_{\min}$ be the *min*-type operation and let φ_{maj} be the *majority*-type operation both defined on the ordered domain $\{0, 1\}$.

▶ **Proposition 20.** *For every natural number n, there is a CSP instance $\mathcal{I}_n$ such that $\mathcal{I}_n$ has a strong $\oplus(\Delta(\{\varphi_{\min}, \varphi_{\text{maj}}\}))$-backdoor set of size zero but every strong $\mathrm{CSP}(\{\varphi_{\min}, \varphi_{\text{maj}}\})$-backdoor set of $\mathcal{I}_n$ has size at least n.*

Proof. Let $\mathrm{MAJ}[a, b, c]$ be the constraint with scope (a, b, c) and whose relation contains all possible tuples except for the tuple $(1, 1, 1)$. Then $\mathrm{MAJ}[a, b, c]$ is not closed under φ_{maj} but it is closed under $\varphi_{\min}$. Similarly, let $\mathrm{MIN}[a, b]$ be the constraint with scope (a, b) and whose relation contains the tuples $(0, 1)$ and $(1, 0)$. Then $\mathrm{MIN}[a, b]$ is not closed under $\varphi_{\min}$ but it is closed under φ_{maj}. We claim that the CSP instance $\mathcal{I}_n$ with variables $\{y_1, \ldots, y_{3n}\} \cup \{z_1, \ldots, z_{2n}\}$ and constraints $\mathrm{MAJ}_i = \mathrm{MAJ}[y_{3i+1}, y_{3i+2}, y_{3i+3}]$ and $\mathrm{MIN}_i = \mathrm{MIN}[z_{2i+1}, z_{2i+2}]$ for every i with $0 \leq i \leq n - 1$, satisfies the claim of the proposition. Towards showing this first note that the empty set is a strong $\oplus(\Delta((\{\varphi_{\min}, \varphi_{\text{maj}}\}))$-backdoor set of $\mathcal{I}_n$ of size 0. This is because $\mathcal{I}_n$ is the disjoint union of variable disjoint constraints, which are either closed under $\varphi_{\min}$ or under φ_{maj}. However it is straightforward to verify that every strong $\mathrm{CSP}(\{\varphi_{\min}, \varphi_{\text{maj}}\})$-backdoor set of $\mathcal{I}_n$ has to either contain at least one variable from every constraint MIN_i or at least one variable from every constraint MAJ_i. Hence the size of any strong $\mathrm{CSP}(\{\varphi_{\min}, \varphi_{\text{maj}}\})$-backdoor set is at least n. ◀

Below we will present an algorithm that detects scattered base classes. For this algorithm it is convenient to deal only with constraint languages that are closed under assignments (as we will define next). This has the advantage that there is no danger that a partially constructed backdoor set becomes invalidated by adding another variable to the backdoor set. In fact, in the orginal paper which introduced backdoor sets [55], this property is even part of the definition. We will also assume that the considered constraint languages contain a redundant constraint which can be used within the algorithm to artificially connect parts of the instance.

A constraint language Γ is *closed under assignments* if for every $C = (S, R)$ such that $R \in \Gamma$ and every assignment α, it holds that $R' \in \Gamma$ where $C|_\alpha = (S', R')$. The lemma below shows that languages closed under assignments are closely related to semi-conservative languages. For a constraint language Γ over a domain $\mathcal{D}$ we denote by Γ^* the smallest constraint language over $\mathcal{D}$ that contains $\Gamma \cup \{\mathcal{D}^2\}$ and is closed under assignments; notice that Γ^* is uniquely determined by Γ. For a set Δ of constraint languages we denote by Δ^* the set $\{\, \Gamma^* : \Gamma \in \Delta \,\}$

▶ **Lemma 21.** *If Δ is a set of globally tractable semi-conservative constraint languages, then $\oplus(\Delta^*)$ is also globally tractable.*

Proof. Evidently, if a semi-conservative language Γ is globally tractable, then so is Γ^*: first, all constraints of the form $(S, \mathcal{D}^2|_\alpha)$ can be detected in polynomial time and removed from the instance without changing the solution, and then each constraint $C' = (S', R')$ with $R' \in \Gamma^* \setminus \Gamma$ can be expressed in terms of the conjunction of a constraint $C = (S, R)$ with $R \in \Gamma$ and unary constraints over variables in $\mathrm{var}(C) \setminus \mathrm{var}(C')$. Now, if each $\Gamma^* \in \Delta^*$ is globally tractable, also $\oplus(\Delta^*)$ is globally tractable, as we can solve connected components independently. ◀

The main technical result for scattered base classes is the next lemma.

▶ **Theorem 22.** *Let Δ be a finite set of finite constraint languages. Then there is an FPT-algorithm that, given a CSP instance $\mathcal{I}$ and a parameter k, either finds a strong $\mathrm{CSP}(\oplus(\Delta^*))$-backdoor set of size at most k or correctly decides that none exists.*

We sketch the main ingredients of the algorithm and refer for the ArXiv version of the original paper [26] for details.

The algorithm uses the technique of iterative compression [50] to transform the problem into a structured subproblem. In this technique, the idea is to start with a sub-instance and a trivial solution for this sub-instance and iteratively expand the sub-instances while compressing the solutions till we solve the problem on the original instance. Specifically, for backdoor detection we are given additional information about the desired solution in the input: we receive an 'old' strong backdoor set which is slightly bigger than our target size, along with information about how this old backdoor set interacts with our target solution.

Further more, the algorithm considers only solutions for instances of the iterative compression problem which have a certain inseparability property and uses an FPT algorithm to test for the presence of such solutions. To be more precise, the algorithm only looks for solutions which leave the omitted part of the old strong backdoor set in a single connected component. Interestingly, even this base case requires the extension of state of the art separator techniques to a CSP setting.

Finally, the general instances of the iterative compression problem are handled using a new pattern replacement technique, which is somehow similar to the protrusion replacement technique [3] but allows the preservation of a much larger set of structural properties (such as containment of disconnected forbidden structures and connectivity across the boundary). This pattern replacement procedure is interleaved with the technique of important separator sequences [44] as well as the above algorithm for inseparable instances.

▶ **Corollary 23.** *Let Δ be a finite set of globally tractable semi-conservative finite constraint languages. Then $\mathrm{CSP}(\oplus(\Delta))$ is fixed-parameter tractable parameterized by the backdoor set size.*

Proof. Let $\mathcal{I}$ be the given CSP instance over domain D and k a parameter such that $\mathcal{I}$ has a strong $\mathrm{CSP}(\oplus(\Delta))$-backdoor set of size $\leq k$. Since $\mathrm{CSP}(\oplus(\Delta)) \subseteq \mathrm{CSP}(\oplus(\Delta^*))$, $\mathcal{I}$ has also a strong $\mathrm{CSP}(\oplus(\Delta^*))$-backdoor set of size $\leq k$, and we compute this backdoor set using Theorem 22. Because of Lemma 21, $\mathrm{CSP}(\oplus(\Delta^*))$ is globally tractable. Hence we can use the backdoor set by solving all the instances that arise by instantiating the backdoor set variables. As the considered languages are finite, they are over a finite domain D, and so the number of tractable instances to consider is at most $|D|^k$. ◀

It might be possible to generalize Corollary 23 to sets of constraint languages characterized by tractable polymorphism predicates. Since the algorithm is already quite complicated for the finite case, this has not yet been checked in detail.

5.1 Extension to Valued CSP

Valued CSP (or VCSP for short) is a powerful framework that entails among others the problems CSP and MAX-CSP as special cases [56]. A VCSP instance consists of a finite set of cost functions over a finite set of variables which range over a domain D, and the task is to find an instantiation of these variables that minimizes the sum of the cost functions. The VCSP framework is robust and has been studied in different contexts in computer science. In its full generality, VCSP considers cost functions that can take as values the rational numbers and positive infinity. CSP (feasibility) and Max-CSP (optimization) arise as special cases by limiting the values of cost functions to $\{0, \infty\}$ and $\{0, 1\}$, respectively. Clearly VCSP is in general intractable. Over the last decades much research has been devoted into the identification of tractable VCSP subproblems. An important line of this research (see, e.g., [36, 39, 54]) is the characterization of tractable VCSPs in terms of restrictions on the underlying *valued constraint language* Γ, i.e., a set Γ of cost functions that guarantees polynomial-time solvability of all VCSP instances that use only cost functions from Γ. The VCSP restricted to instances with cost functions from Γ is denoted by VCSP[Γ].

The definitions of strong backdoor sets, scattered base classes, etc., generalize straightforwardly from the CSP setting to VCSP, and we omit the details. A valued constraint language is *conservative* if it contains all unary cost functions [39].

Theorem 23 generalizes to VCSP in the following way:

▶ **Theorem 24.** *Let Δ be a finite set of globally tractable conservative valued constraint languages of bounded domain size and bounded arity. Then* VCSP($\oplus(\Delta)$) *is fixed-parameter tractable parameterized by the backdoor set size.*

Here it is important to note that a valued constraint language of bounded domain and arity can be infinite. Hence the technique used for establishing Theorem 23 does not directly apply. However, it turns out that one can transform the backdoor set detection problem from a general scattered class VCSP($\oplus(\Delta)$) to a scattered class VCSP($\oplus(\Delta')$) over a finite set Δ' of *finite* valued constraint languages. The reduction does not preserve VCSP solutions, but preserves backdoor sets. Once the backdoor set is found, one applies it to the original VCSP instance.

6 Backdoors of Small Treewidth

In this section we outline a new concept that allows us to algorithmically exploit a backdoor set even if it is large, as long as it induces a graph with a sufficiently simple structure. More specifically, we associate with the backdoor set a certain *torso graph* and measure its structure in terms of the widely used graph invariant treewidth. Minimizing this treewidth over all strong backdoor sets X of a CSP instance $\mathcal{I}$ into CSP(Γ) (for a fixed constraint language Γ) gives as a new "hybrid" parameter, the *backdoor-treewidth*.

Let $\mathcal{I}$ be a CSP instance and X a subset of its variables. We define the *torso graph* of $\mathcal{I}$ with respect to X, denoted $\mathsf{torso}_{\mathcal{I}}(X)$, as follows. The vertex set of $\mathsf{torso}_{\mathcal{I}}(X)$ is X, and the graph contains an edge $\{x, y\}$ if x and y appear together in the scope of a constraint or x and y are in the scopes of constraints that belong to the same connected component of $\mathcal{I} - X$ (see Section 5). Here $\mathcal{I} - X$ denotes the CSP instance obtained from $\mathcal{I}$ by deleting the variables $x \in X$ from all constraint scopes and deleting the corresponding entries from the constraint relations.

Let $G = (V, E)$ be a graph. A tree decomposition of G is a pair $(T, \mathcal{X} = \{X_t\}_{t \in V(T)})$ where T is a tree and $\mathcal{X}$ is a collection of subsets of V such that: (i) for each edge $\{u, v\} \in E$ there exists a node t of T such that $\{u, v\} \subseteq X_t$, and (ii) for each $v \in V$, the set $\{t \mid v \in X_t\}$ induces in T a nonempty connected subtree. The width of $(T, \mathcal{X})$ is equal to $\max\{|X_t| - 1 \mid t \in V(T)\}$ and the *treewidth* of G, denoted $\mathsf{tw}(G)$, is the minimum width over all tree decompositions of G.

Let $\mathcal{I}$ be a CSP instance and X a strong backdoor set of $\mathcal{I}$ into CSP(Γ). The *width* of X is the treewidth of the torso graph $\mathsf{torso}_{\mathcal{I}}(X)$, and the *backdoor-treewidth* of $\mathcal{I}$ with respect to Γ is the smallest width over all strong backdoor sets X of $\mathcal{I}$ into CSP(Γ).

▶ **Theorem 25.** *For each finite constraint language Γ, there is an FPT-algorithm that, given a CSP instance $\mathcal{I}$ and a parameter k, either finds a strong* CSP(Γ)*-backdoor set of with at most k or correctly decides that none exists.*

The proof of the theorem makes use of a new notion of "boundaried CSP instances" defined in the spirit of boundaried graphs, and uses a new replacement framework inspired by the graph replacement tools dating back to the results of Fellows and Langston [20], combined with the so-called recursive-understanding technique [31]

Once a backdoor set of small with is found, one can apply standard dynamic programming techniques to solve CSP and #CSP.

▶ **Corollary 26.** *For each finite (#-)tractable constraint language Γ,* CSP *(or* #CSP*, respectively) is fixed-parameter tractable parameterized by the backdoor treewidth with respect to Γ.*

7 Related Work

Related work on backdoor sets for CSP includes a paper by Bessière et at. [2] who consider so-called *partition backdoor sets* which are less general than the backdoor sets we considered in Section 4 (see [28, Section 7]); they also provide some initial empirical results which show that this concept has practical potential. A further work on CSP backdoor sets is a paper by Carbonnel et al. [7] who show W[2]-hardness for strong backdoor set detection when parameterized by the size of the backdoor set, even for CSP-instances with only one constraint (however with unbounded domain and unbounded arity). They also give a fixed-parameter algorithm for strong backdoor set detection parameterized by the size of the backdoor set and the maximum arity of any constraint, if the base class is "h-Helly" for a fixed integer h and under the additional assumption that the domain is a finite subset of the natural numbers, which comes with a fixed ordering (see also the recent survey by Carbonnel and Cooper [8]).

As mentioned at the beginning of this survey, there is much work on backdoor sets in the context of SAT, most of it is covered in the survey paper [29]. More recent additions include the detection of strong backdoor sets with respect to the base class of CNF formulas whose incidence graph is of bounded treewidth [24, 30]. We would like to note that in SAT the application of a partial assignment to a CNF formula results in the deletion of all satisfied clauses, which provides an additional power for strong backdoor sets, and provides an additional challenge for their detection.

The parameterized complexity of finding and using backdoor sets has been studied for several problems besides SAT and CSP, including the satisfiability of quantified Boolean formulas [51], disjunctive answer set programming [21, 22], abductive reasoning [49], abstract argumentation [15], planning [41, 42], and linear temporal logic [46].

Finally, we would like to mention that in the area of graph algorithms the notion of *modulators* [6] is closely related to the concept of backdoor sets. A modulator of a graph G into a fixed graph class $\mathcal{C}$ is a set M of vertices of G such that deleting M from G moves G into the class $\mathcal{C}$. By considering modulators into graph classes $\mathcal{C}$ where certain NP-hard graph problems can be solved in polynomial time, one can often lift the tractability of the problem to fixed-parameter tractability for general graphs, parameterized by the size of the modulator. Therefore the fixed-parameter tractability of the detection of modulators (parameterized by modulator size) is of interest. Important results include modulators to bipartite graphs [43, 50], to chordal graphs [45], and to forests [12]. Recently alternative parameterizations of modulators that are more general than their size have been explored [16, 17].

8 Conclusion

We presented parameterized complexity results on CSP backdoor sets into bases classes defined via restrictions on the constraint languages. The presented results show that the notion of CSP backdoor sets provides an interesting area of research, which on one side builds upon and extends classical CSP-tractability results (in form of the considered base classes), and on the other side uses advanced algorithmic methods from the area of parameterized complexity. There are plenty of questions that are mostly unexplored, such as CSP backdoor sets into base classes defined by structural properties, base classes defined by "hybrid" concepts like forbidden patterns [10], or CSP backdoor sets for instances with global constraints. Another promising direction of future research is to parameterize backdoor sets not by their size but by structural properties of the backdoor set and how it interacts with the rest of the instance, similar to the parameters that have been considered for modulators [16, 17]; we have outlined first results into this direction in Section 6. We hope that this survey stimulates further research on CSP backdoor sets.

References

1 Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. of the ACM*, 61(1):3:1–3:19, 2014.

2 Christian Bessiére, Clément Carbonnel, Emmanuel Hebrard, George Katsirelos, and Toby Walsh. Detecting and exploiting subproblem tractability. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013.

3 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) kernelization. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 629–638. IEEE Computer Soc., Los Alamitos, CA, 2009.

4 Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. of the ACM*, 53(1):66–120, 2006.

5 Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

6 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.

7 Clément Carbonnel, Martin C. Cooper, and Emmanuel Hebrard. On backdoors to tractable constraint languages. In *Principles and Practice of Constraint Programming – 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings*, volume 8656 of *Lecture Notes in Computer Science*, pages 224–239. Springer Verlag, 2014.

8 Clément Carbonnel and Martin C. Cooper. Tractability in constraint satisfaction problems: a survey. *Constraints*, pages 1–30, 2015. doi:10.1007/s10601-015-9198-6.

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

10 David A. Cohen, Martin C. Cooper, Páidí Creed, Dániel Marx, and András Z. Salamon. The tractability of CSP classes defined by forbidden patterns. *J. Artif. Intell. Res.*, 45:47–78, 2012.

11 Nadia Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *J. of Computer and System Sciences*, 51(3):511–522, 1995.

12 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.23.

13 R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, New York, 1999.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

15 Wolfgang Dvorák, Sebastian Ordyniak, and Stefan Szeider. Augmenting tractable fragments of abstract argumentation. *Artificial Intelligence*, 186:157–173, 2012.

16 Eduard Eiben, Robert Ganian, and Stefan Szeider. Meta-kernelization using well-structured modulators. In Thore Husfeldt and Iyad A. Kanj, editors, *Parameterized and Exact Computation – 10th International Symposium, IPEC 2014, Patras, Greece, September 16-18, 2015. Revised Selected Papers*, volume 43 of *LIPIcs*, pages 114–126, 2015. doi:10.4230/LIPIcs.IPEC.2015.114.

17 Eduard Eiben, Robert Ganian, and Stefan Szeider. Solving problems on graphs of high rank-width. In *Algorithms and Data Structures Symposium (WADS 2015), August 5-7, 2015, University of Victoria, BC, Canada*, LNCS, pages 314–326. Springer Verlag, 2015. URL: http://arxiv.org/abs/1507.05463, doi:10.1007/978-3-319-21840-3_26.

18 Tomás Feder and Moshe Y. Vardi. Monotone monadic snp and constraint satisfaction. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 612–622. ACM, 1993.

19 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

20 Michael R. Fellows and Michael A. Langston. An analogue of the myhill-nerode theorem and its use in computing finite-basis characterizations (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October – 1 November 1989*, pages 520–525. IEEE Computer Society, 1989.

21 Johannes Klaus Fichte and Stefan Szeider. Backdoors to normality for disjunctive logic programs. *ACM Trans. Comput. Log.*, 17(1), 2015. doi:10.1145/2818646.

22 Johannes Klaus Fichte and Stefan Szeider. Backdoors to tractable answer set programming. *Artificial Intelligence*, 220:64–103, March 2015. doi:10.1016/j.artint.2014.12.001.

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

24 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, M. S. Ramanujan, and Saket Saurabh. Solving *d*-SAT via backdoors to small treewidth. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 630–641, 2015.

25 Robert Ganian, M. S. Ramanujan, and Stefan Szeider. Backdoors to valued constraint satisfaction. In *Proceedings of CP 2016, the 22nd International Conference on Principles and Practice of Constraint Programming Toulouse, France 5-9 September 2016*, Lecture Notes in Computer Science. Springer Verlag, 2016. to appear.

26 Robert Ganian, M. S. Ramanujan, and Stefan Szeider. Discovering archipelagos of tractability for constraint satisfaction and counting. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1670–1681, 2016. Full version to appear in the *ACM Transactions on Algorithms*. doi:10.1137/1.9781611974331.ch114.

27 Robert Ganian, M. S. Ramanujan, and Stefan Szeider. Combining treewidth and backdoors for CSP. In *Proceedings of STACS 2017, the 34th International Symposium on Theoretical Aspects of Computer Science*, 2017. To appear. URL: https://arxiv.org/abs/1610.03298.

28 Serge Gaspers, Neeldhara Misra, Sebastian Ordyniak, Stefan Szeider, and Stanislav Živný. Backdoors into heterogeneous classes of SAT and CSP. *J. of Computer and System Sciences*, 2015. To appear, a preliminary version is available from https://arxiv.org/abs/1509.05725.

29 Serge Gaspers and Stefan Szeider. Backdoors to satisfaction. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel 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.

30 Serge Gaspers and Stefan Szeider. Strong backdoors to bounded treewidth SAT. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 489–498. IEEE Computer Society, 2013.

31 Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 479–488. ACM, 2011.

32 Pavol Hell and Jaroslav Nesetril. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.

33 Pawel M. Idziak, Petar Markovic, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.

34 Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *J. of the ACM*, 44(4):527–548, 1997.

35 Peter Jeavons, David A. Cohen, and Martin C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

36 Peter Jeavons, Andrei A. Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bulletin of the European Association for Theoretical Computer Science*, 113, 2014.

37 Phokion G. Kolaitis. Constraint satisfaction, databases, and logic. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 1587–1595. Morgan Kaufmann, 2003.

38 Phokion G. Kolaitis and Moshe Y. Vardi. A logical approach to constraint satisfaction. In *Finite model theory and its applications*, Texts Theoret. Comput. Sci. EATCS Ser., pages 339–370. Springer Verlag, 2007.

39 Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. *J. of the ACM*, 60(2):Art. 10, 38, 2013.

40 Marcin Kozik, Andrei Krokhin, Matt Valeriote, and Ross Willard. Characterizations of several Maltsev Conditions. *Algebra Universalis*, 73(3-4):205–224, 2015.

41 Martin Kronegger, Sebastian Ordyniak, and Andreas Pfandler. Backdoors to planning. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27-31, 2014, Québec City, Québec, Canada.*, pages 2300–2307. AAAI Press, 2014.

42 Martin Kronegger, Sebastian Ordyniak, and Andreas Pfandler. Variable-deletion backdoors to planning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 3305–3312. AAAI Press, 2015.

43 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014. doi:10.1145/2566616.

44 Daniel Lokshtanov and M. S. Ramanujan. Parameterized tractability of multiway cut with parity constraints. In *ICALP 2012, Automata, languages, and programming. Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 750–761. Springer Verlag, 2012.

45 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

46 Arne Meier, Sebastian Ordyniak, M. S. Ramanujan, and Irena Schindler. Backdoors for linear temporal logic. In *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, Leibniz International Proceedings in Informatics (LIPIcs), pages 23:1–23:17. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.23.

47 Ugo Montanari. Networks of constraints: fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

48 Naomi Nishimura, Prabhakar Ragde, and Stefan 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.

49 Andreas Pfandler, Stefan Rümmele, and Stefan Szeider. Backdoors to abduction. In *Proceedings of IJCAI 2013, the 23th International Joint Conference on Artificial Intelligence, August 3-9, 2013, Beijing, China*, 2013.

50 Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

51 Marko Samer and Stefan Szeider. Backdoor sets of quantified Boolean formulas. *Journal of Automated Reasoning*, 42(1):77–97, 2009. URL: https://www.ac.tuwien.ac.at/files/pub/SamerSzeider09a.pdf.

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

53 Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 695–704. ACM, 2013.

54 Johan Thapper and Stanislav Živný. Necessary conditions for tractability of valued CSPs. *SIAM J. Discrete Math.*, 29(4):2361–2384, 2015.

55 Ryan Williams, Carla Gomes, and Bart Selman. Backdoors to typical case complexity. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI 2003*, pages 1173–1178. Morgan Kaufmann, 2003.

56 Stanislav Živný. *The complexity of valued constraint satisfaction problems.* Cognitive Technologies. Springer Verlag, 2012.