

# Backdoors to Planning

**Martin Kronegger**

Vienna University of Technology,  
Vienna, Austria  
kronegger@dbai.tuwien.ac.at

**Sebastian Ordyniak**

Masaryk University,  
Brno, Czech Republic  
sordyniak@gmail.com

**Andreas Pfandler**

Vienna University of Technology,  
Vienna, Austria  
pfandler@dbai.tuwien.ac.at

## Abstract

Backdoors measure the distance to tractable fragments and have become an important tool to find fixed-parameter tractable (fpt) algorithms. Despite their success, backdoors have not been used for planning, a central problem in AI that has a high computational complexity. In this work, we introduce two notions of backdoors building upon the causal graph. We analyze the complexity of finding a small backdoor (detection) and using the backdoor to solve the problem (evaluation) in the light of planning with (un)bounded plan length/domain of the variables. For each setting we present either an fpt-result or rule out the existence thereof by showing parameterized intractability. In three cases we achieve the most desirable outcome: detection and evaluation are fpt.

## 1 Introduction

Planning is one of the central formalisms in AI. Unfortunately, the expressive power of planning comes at the cost of high computational complexity. In general, already propositional STRIPS planning is PSPACE-complete for unbounded plan length. In order to cope with this high complexity, several fragments of planning have been considered where planning becomes tractable. For propositional STRIPS a comprehensive complexity analysis was performed by Bylander (1994). Later, Bäckström and Nebel (1995) presented a similar analysis for the SAS<sup>+</sup> formalism, where the state variables range over multi-valued domains.

A more fine-grained understanding of the hardness of a problem can be obtained from the viewpoint of parameterized complexity theory (Downey and Fellows 1999). Here one is interested in identifying one or multiple features of the instance – the so-called parameters – which capture the combinatorial explosion. More formally, the time needed to solve the problem is not only measured in terms of the input size  $n$ , but also depends on the parameter (or combination of parameters)  $k$ . The class of efficiently solvable problems is FPT (*fixed-parameter tractable*), i.e., the problem can be solved by an fpt-algorithm in time  $f(k) \cdot n^{\mathcal{O}(1)}$ , where  $n$  denotes the size of the problem instance and  $f(k)$  is a computable function depending on parameter  $k$  only (and not on  $n$ ). The exponential time complexity is thus confined to the

parameter, i.e., the function  $f(k)$ . Therefore, an fpt-algorithm can be considered efficient as long as the parameter values of an instance are sufficiently low. Furthermore notice that an fpt-result immediately yields tractability of the problem if parameter  $k$  is bounded by a constant. Techniques from parameterized complexity have been successfully used to tackle hard problems, e.g., in Knowledge Representation & Reasoning (Gottlob and Szeider 2008). For planning a parameterized complexity analysis was initiated by Downey, Fellows, and Stege (1999). In recent works by Bäckström et al. (2012) and Kronegger, Pfandler, and Pichler (2013), several fpt-results have been obtained for SAS<sup>+</sup> and propositional STRIPS. Despite this success the desire for additional fpt-algorithms remains.

A powerful tool to obtain fpt-algorithms are so-called *backdoors* (for a survey see Gaspers and Szeider (2012b)). Backdoors were originally introduced by Williams, Gomes, and Selman (2003) to explain the behavior of SAT and CSP solvers on practical instances. The basic idea is that the size of a backdoor set measures the distance to a tractable fragment of the problem. In the past, the backdoor approach has been used to obtain fpt-results for SAT (Nishimura, Ragde, and Szeider 2004; Gaspers and Szeider 2012a), QBFs (Samer and Szeider 2009), ASP (Fichte and Szeider 2011a), and Argumentation (Dvorák, Ordyniak, and Szeider 2012). Furthermore, several experiments regarding the size of the backdoors of SAT and ASP instances were performed (Kilby et al. 2005; Gregory, Fox, and Long 2008; Fichte and Szeider 2011b). Interestingly, some of the instances were obtained through SAT- and ASP-encodings of planning instances. Recently, backdoors have also been used to construct parameterized reductions to SAT for problems harder than NP such as ASP (Fichte and Szeider 2013) and Abduction (Pfandler, Rümmele, and Szeider 2013). However, backdoors for planning have not been considered yet.

In contrast to other problems such as SAT, where backdoors can be defined in a direct way, the situation is different for planning. Here, an additional layer is needed. The challenge is to find a suitable notion capturing the underlying structure of the planning instance, which in the next step can be used to define the desired notion of backdoors.

## Main contributions

In this work, we introduce two natural notions of backdoors for planning, which are based on the underlying *causal graph*

of the planning instance.

For both of the new notions of backdoors, we perform a comprehensive parameterized complexity analysis (see Table 1 for an overview). In more detail, we analyze the complexity of the two phases of the backdoor approach: (i) finding a small backdoor (*detection phase*), and (ii) using the additional information given in the backdoor to solve the planning instance (*evaluation phase*). In the evaluation phase, we additionally consider a bound on the plan length and/or on the domain of the variables as additional restrictions. For each of the above settings we present either an  $\text{fpt}$ -result, or show parameterized intractability. This gives us a *complete* picture of the parameterized complexity of the backdoor approach w.r.t. the considered notions of backdoors.

Among our  $\text{fpt}$ -results is the first  $\text{fpt}$ -algorithm for planning with unbounded plan length that does neither limit the number of variables nor actions in the planning instance.

## 2 Preliminaries

We assume the reader to be familiar with the basics of complexity theory and planning. For an introduction to complexity theory we refer, e.g., to the book of Papadimitriou (1994). For  $n \in \mathbb{N}$ , we use  $[n]$  to denote the set  $\{1, \dots, n\}$ .

**Parameterized Complexity.** Parameterized algorithmics (cf. Downey and Fellows (1999), Flum and Grohe (2006), Niedermeier (2006)) is a promising approach to obtain efficient algorithms for fragments of intractable problems. In a parameterized complexity analysis the runtime of an algorithm is studied with respect to a parameter  $k \in \mathbb{N}$  and input size  $n$ . The basic idea is to find a parameter that describes the structure of the instance such that the combinatorial explosion can be confined to this parameter. The most favorable class is  $\text{FPT}$  (*fixed-parameter tractable*) which contains all problems that can be decided by an algorithm running in time  $f(k) \cdot n^{\mathcal{O}(1)}$ , where  $f$  is a computable function. We call such an algorithm fixed-parameter tractable ( $\text{fpt}$ ).

Formally, a *parameterized problem* is a subset of  $\Sigma^* \times \mathbb{N}$ , where  $\Sigma$  is the input alphabet. Problem reductions now also have to take the parameter into account. Let  $L_1$  and  $L_2$  be parameterized problems, with  $L_1 \subseteq \Sigma_1^* \times \mathbb{N}$  and  $L_2 \subseteq \Sigma_2^* \times \mathbb{N}$ . A *parameterized reduction* (or  $\text{fpt}$ -reduction) from  $L_1$  to  $L_2$  is a mapping  $P : \Sigma_1^* \times \mathbb{N} \rightarrow \Sigma_2^* \times \mathbb{N}$  such that (i)  $(x, k) \in L_1$  iff  $P(x, k) \in L_2$ ; (ii) the mapping can be computed by an  $\text{fpt}$ -algorithm w.r.t. parameter  $k$ ; (iii) there is a computable function  $g$  such that  $k' \leq g(k)$ , where  $(x', k') = P(x, k)$ .

Next, we will define the classes capturing fixed-parameter intractability needed in this work. For further details we have to refer to the literature on parameterized complexity theory.

The class  $\text{W}[1]$  contains all problems that are  $\text{fpt}$ -reducible to  $\text{INDEPENDENT SET}$  when parameterized by the size of the solution, i.e., the independent set (Downey and Fellows 1999; Flum and Grohe 2006). The class  $\text{paraNP}$  (Flum and Grohe 2003) is defined as the class of problems that are solvable by a non-deterministic Turing-machine in  $\text{fpt}$ -time. In our  $\text{paraNP}$ -hardness proofs, we will make use of the following characterization of  $\text{paraNP}$ -hardness given by Flum and Grohe (2006), Theorem 2.14: any parameterized problem that remains  $\text{NP}$ -hard when the parameter is set to some constant

is  $\text{paraNP}$ -hard. The following relations between the parameterized complexity classes hold:  $\text{FPT} \subseteq \text{W}[1] \subseteq \text{paraNP}$ . Showing  $\text{W}[1]$ -hardness for a problem rules out the existence of a fixed-parameter algorithm under the usual complexity theoretic assumption  $\text{FPT} \neq \text{W}[1]$ .

**Planning.** Let  $V = \{v_1, \dots, v_n\}$  be a finite set of *variables* over a finite *domain*  $D$ . Implicitly define  $D^+ = D \cup \{\mathbf{u}\}$ , where  $\mathbf{u}$  is a special “undefined” value not present in  $D$ . Then  $D^n$  is the set of *total states* and  $(D^+)^n$  is the set of *partial states* over  $V$  and  $D$ . Clearly,  $D^n \subseteq (D^+)^n$ . The value of a variable  $v$  in a state  $s \in (D^+)^n$  is denoted by  $s[v]$ . A  $\text{SAS}^+$  *instance* is a tuple  $\mathbb{P} = \langle V, D, A, I, G \rangle$  where  $V$  is a set of variables,  $D$  is a domain,  $A$  is a set of *actions*,  $I \in D^n$  is the *initial state* and  $G \in (D^+)^n$  is the (partial) *goal state*. Each action  $a \in A$  has a *precondition*  $\text{pre}(a) \in (D^+)^n$  and an *effect*  $\text{eff}(a) \in (D^+)^n$ . We will frequently use the convention that a variable has value  $\mathbf{u}$  in a precondition/effect unless a value is explicitly specified. Let  $a \in A$  and let  $s \in D^n$ . Then  $a$  is *valid* in  $s$  if for all  $v \in V$ , either  $\text{pre}(a)[v] = s[v]$  or  $\text{pre}(a)[v] = \mathbf{u}$ . Furthermore, the *result of  $a$  in  $s$*  is a state  $t \in D^n$  defined such that for all  $v \in V$ ,  $t[v] = \text{eff}(a)[v]$  if  $\text{eff}(a)[v] \neq \mathbf{u}$  and  $t[v] = s[v]$  otherwise.

Let  $s_0, s_\ell \in D^n$  and let  $\omega = \langle a_1, \dots, a_\ell \rangle$  be a sequence of actions (of length  $\ell$ ). Then  $\omega$  is a *plan from  $s_0$  to  $s_\ell$*  if either (i)  $\omega = \langle \rangle$  and  $\ell = 0$ , or (ii) there are states  $s_1, \dots, s_{\ell-1} \in D^n$  such that for all  $1 \leq i \leq \ell$ ,  $a_i$  is valid in  $s_{i-1}$  and  $s_i$  is the result of  $a_i$  in  $s_{i-1}$ . A state  $s \in D^n$  is a *goal state* if for all  $v \in V$ , either  $G[v] = s[v]$  or  $G[v] = \mathbf{u}$ . An action sequence  $\omega$  is a *plan for  $\mathbb{P}$*  if  $\omega$  is a plan from  $I$  to some goal state. We will study backdoors for the following problems:

**SAS<sup>+</sup> PLANNING**

*Instance:* A  $\text{SAS}^+$  instance  $\mathbb{P}$ .

*Question:* Does  $\mathbb{P}$  have a plan?

**BOUNDED SAS<sup>+</sup> PLANNING**

*Instance:* A  $\text{SAS}^+$  instance  $\mathbb{P}$  and a positive integer  $k$ .

*Parameter:*  $k$ .

*Question:* Does  $\mathbb{P}$  have a plan of length at most  $k$ ?

Notice that the propositional version of the well-known STRIPS planning language is a special case of  $\text{SAS}^+$ .

Let  $\mathbb{P} = \langle V, D, A, I, G \rangle$  be an  $\text{SAS}^+$  instance,  $V' \subseteq V$ , and  $A' \subseteq A$ . Then we denote by  $\mathbb{P}[V']$  the  $\text{SAS}^+$  instance  $\langle V', D_r, A_r, I_r, G_r \rangle$ , where  $D_r$  is the restriction of  $D$  to the domains of the variables in  $V'$ ,  $A_r$  are the actions in  $A$  whose preconditions and effects are restricted to the variables in  $V'$ , and  $I_r$  and  $G_r$  are the restriction of  $I$  and  $G$  to the variables in  $V'$ . We write  $\mathbb{P} \setminus V'$  for the instance  $\mathbb{P}[V \setminus V']$ . Similarly, we denote by  $\mathbb{P}[A']$  the  $\text{SAS}^+$  instance  $\langle V, D, A', I, G \rangle$  and by  $\mathbb{P} \setminus A'$  the  $\text{SAS}^+$  instance  $\mathbb{P}[A \setminus A']$ .

## 3 Using the Causal Graph for Backdoors

In this work we will introduce two new types of backdoors. Before we start with the presentation of the details, we give a high-level introduction to the backdoor approach.

The backdoor approach can be separated into two phases. In the first phase (*detection*) one searches for a set, i.e., the backdoor, whose size measures the distance of the given instance to a tractable base class. In the second phase (*evaluation*) one makes use of the information of the backdoor to

Setting	Detection		Evaluation	
	(un)bounded domain	(un)bounded plan length	bounded plan length	unbounded plan length
variable-deletion	(un)bounded domain	in FPT (Thm. 3)	W[1]-hard (Thm. 4)	W[1]-hard (Thm. 4)
action-deletion	bounded domain	in FPT (Thm. 7)	in FPT (Thm. 10)	in FPT (Thm. 9)
	unbounded domain	in FPT (Thm. 7)	in FPT (Thm. 10)	paraNP-hard (Thm. 8)

Table 1: Complexity map of backdoors to planning.

solve the problem. Usually, the size of the backdoor is considered as parameter in the detection and evaluation problem. Hence, if both problems are fpt, they can be solved efficiently as long as the backdoor is of moderate size.

For instance, for the SAT problem, one searches for a small set of variables of size  $k$  such that the given formula can be reduced to  $2^k$  easy formulas that belong to the desired tractable base class (e.g., Horn or Krom). For planning, however, it is not immediately clear how to break a hard instance into multiple easy instances by using a backdoor set.

The basic idea of this work is to use some underlying structure of the planning instance, namely the *causal graph*, instead of the planning instance itself to define the backdoor. Therefore, we first need to recapitulate the concept of the causal graph (Knoblock 1994; Brafman and Domshlak 2006; Chen and Giménez 2010; Bäckström and Jonsson 2013).

The *causal graph*  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$  of a SAS<sup>+</sup> instance  $\mathbb{P} = \langle V, D, A, I, G \rangle$  is the directed graph with vertices  $V$  that contains an arc  $(v, v')$ , for distinct  $v, v' \in V$ , if either there is an action  $a \in A$  with  $\text{pre}(a)[v] \neq \mathbf{u}$  and  $\text{eff}(a)[v'] \neq \mathbf{u}$  or there is an action  $a \in A$  with  $\text{eff}(a)[v] \neq \mathbf{u}$  and  $\text{eff}(a)[v'] \neq \mathbf{u}$ . If  $C$  is a set of vertices of a subgraph of  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$  we denote by  $\mathbb{P}[C]$  the SAS<sup>+</sup> instance  $\mathbb{P}[V']$  where  $V' \subseteq V$  are all variables that correspond to vertices of  $C$ . We denote by  $\text{cc-size}(H)$  the cardinality of the largest weakly connected component of the directed graph  $H$ . If it is clear from the context, we will write “components” instead of “weakly connected components of the causal graph  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$ ”.

In order to be able to define the backdoors, we need to find a suitable, tractable base class. From the literature it is known that SAS<sup>+</sup> PLANNING is solvable in polynomial time if the maximum component size of the underlying graph can be bounded by a constant  $c$ , i.e.,  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})) \leq c$ .

**Theorem (Chen and Giménez (2010)).** *Let  $c$  be a constant. Then SAS<sup>+</sup> PLANNING can be solved in polynomial time for instances where  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})) \leq c$ .*

For the setting with bounded plan length it is easy to obtain a similar result. It suffices to construct the state transition graph (of size  $\mathcal{O}(|D|^c)$ ) for each component and compute the shortest path to the partial goal. An arbitrary combination of the shortest plans for each component gives a solution.

**Proposition 1.** *Let  $c$  be a constant. Then  $c$ -BOUNDED SAS<sup>+</sup> PLANNING can be solved in polynomial time for instances where  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})) \leq c$ .*

The backdoors introduced in this work measure the distance of the planning instance  $\mathbb{P}$  to a tractable planning instance  $\mathbb{P}'$  that has bounded component size. We will consider two natural ways to decompose the components of the causal graph. The backdoor set  $S$  either contains the *variables* or the *actions* that are removed from the instance to reduce the

size of the weakly connected components. Notice that from the viewpoint of classical complexity theory the evaluation problem remains as hard as the original planning problem, because one is free to add all variables or all actions to the backdoor set.

In this work, each backdoor type is considered in the light of four different settings of planning. We consider the SAS<sup>+</sup> PLANNING and the BOUNDED SAS<sup>+</sup> PLANNING problem in case of an bounded or unbounded domain of the variables. For each case we will present either an fpt-algorithm or hardness for the classes W[1] or paraNP. The results of this work are summarized in Table 1. Due to space limitations several proof details had to be omitted.

### 3.1 Variable-Deletion-Backdoors

In this section we consider planning instances that have a small number of variables whose removal results in a causal graph with components of bounded size. We show that even though the detection of these planning instances is fixed-parameter tractable (Theorem 3) this does not hold for the evaluation problem even for planning instances with bounded domain (Theorem 4).

We start by defining the detection problem for the setting where variables are allowed to be removed (*variable-deletion-backdoors*).

*c*-CAUSAL DETECTION[VARIABLES]

*Instance:* A SAS<sup>+</sup> instance  $\mathbb{P}$  and a positive integer  $k$ .

*Parameter:*  $k$ .

*Question:* Is there a set  $S$  of at most  $k$  variables of  $\mathbb{P}$  such that  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)) \leq c$ ?

Intuitively, a variable-deletion-backdoor captures the distance (in terms of variables that need to be removed) to an instance where all variables can be partitioned into small sets that are relevant for independent subtasks. Observe that the property of having small components in the causal graph is very fragile: Even if the variables form small independent components, adding a single variable that occurs in all actions in the precondition creates a single, big component in the causal graph. Variable-deletion-backdoors can be seen as a more robust notion. In the example above a backdoor of size one (containing the newly introduced variable) is sufficient.

Next, we build upon this backdoor and extend the problems SAS<sup>+</sup> PLANNING and BOUNDED SAS<sup>+</sup> PLANNING to make use of a previously computed backdoor  $S$ .

*c*-CAUSAL EVALUATION[VARIABLES]

*Instance:* A SAS<sup>+</sup> instance  $\mathbb{P}$  and a set  $S$  of variables of  $\mathbb{P}$  such that  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)) \leq c$ .

*Parameter:*  $|S|$ .

*Question:* Does  $\mathbb{P}$  have a plan?

*c*-BOUNDED CAUSAL EVALUATION[VARIABLES]

*Instance:* A SAS<sup>+</sup> instance  $\mathbb{P}$  and a set  $S$  of variables of  $\mathbb{P}$  such that  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)) \leq c$ .

*Parameter:*  $|S| + k$ .

*Question:* Does  $\mathbb{P}$  have a plan of length at most  $k$ ?

We start the analysis by a discussion of the complexity of the detection problem. To justify a parameterized complexity analysis, we first show NP-hardness.

**Theorem 2.** 1-CAUSAL DETECTION[VARIABLES] is NP-hard even for planning instances with bounded domain.

*Proof.* Let  $\langle G' = (V, E), k \rangle$ , with  $|V| = n$ , be an instance of the well-known VERTEX COVER problem. We construct an instance  $\langle \mathbb{P}, k \rangle$  of 1-CAUSAL DETECTION[VARIABLES] in the following way: Let  $D = \{0, 1\}$ ,  $I = 0^n$  and  $G = 1^n$ . The planning instance  $\mathbb{P}$  contains one variable for each vertex in  $V$ . Furthermore, let  $A$  be the union of all actions of the form  $a_{x,y}$  with  $\text{eff}(a_{x,y})[x] = \text{eff}(a_{x,y})[y] = 1$  where  $\{x, y\} \in E$ . Then,  $\mathbb{P} := \langle V, D, A, I, G \rangle$ .

From the construction it is easy to see that every set  $S$  with  $|S| = k$  that is a vertex cover of  $G'$  is a set of  $k$  vertices such that  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)) = 1$  and vice versa.  $\square$

In the next theorem we will show that the detection problem is fixed-parameter tractable when parameterized by the size of the backdoor set  $S$ . To establish this result we will make use of Courcelle's theorem and present an encoding in Monadic Second-Order logic (MSO).

For the proof of the theorem, we need further definitions. Let  $G = (V_G, E_G)$  be a graph. A *tree-decomposition* of  $G$  is a pair  $\langle T, \chi \rangle$  where  $T = (V_T, E_T)$  is a tree and  $\chi$  is a labeling function that maps every vertex  $x \in V_T$  to a set of vertices  $\chi(x) \subseteq V_G$  with the following properties: (i) for every vertex  $y \in V_G$  there is a node  $x \in V_T$  such that  $y \in \chi(x)$ , (ii) for every edge  $\{y_1, y_2\} \in E_G$  there is a node  $x \in V_T$  such that  $y_1, y_2 \in \chi(x)$ , and (iii) for all  $y_1, y_2, y_3 \in V_T$ , if  $y_2$  lies on the unique path from  $y_1$  to  $y_3$  in  $T$ , then  $\chi(y_1) \cap \chi(y_3) \subseteq \chi(y_2)$ . The width of a tree-decomposition  $\langle T, \chi \rangle$  is defined as  $\max_{x \in V_T} \{|\chi(x)|\} - 1$ . The treewidth  $\text{tw}(G)$  of a graph  $G$  is the minimum width over all its tree-decompositions. Bodlaender (1996) showed that for any fixed  $c$ , one can check in linear time (and hence in fpt-time when parameterized by  $c$ ) whether  $\text{tw}(G) \leq c$ .

Monadic Second-Order (MSO) logic is first-order logic extended by set variables. Courcelle (1990) showed that checking whether a graph  $G$  satisfies an MSO formula  $\varphi$  is fixed-parameter tractable when parameterized by the treewidth  $\text{tw}(G)$  and the length of  $\varphi$ .

**Theorem 3.** *c*-CAUSAL DETECTION[VARIABLES] is fpt.

*Proof.* Let  $\langle \mathbb{P}, k \rangle$  be an instance of *c*-CAUSAL DETECTION[VARIABLES] over variables  $V$ . We can construct  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$  in polynomial time. For every YES-instance the following holds:  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$  can be partitioned into  $S$  and  $P_1, \dots, P_m$  such that  $|S| \leq k$ ,  $|P_i| \leq c$  for  $i \in [m]$ , and for every edge  $\{x, y\}$  between two different components either  $x \in S$  or  $y \in S$ . Hence, the  $P_i$  are the remaining weakly connected components if the vertices in  $S$  are removed from  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$ . Therefore, a valid tree-decomposition  $\langle T, \chi \rangle$  of

$\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$  can be constructed in the following way:  $T$  is a complete bipartite graph with vertices  $b_S$  and  $b_1, \dots, b_m$  such that  $b_S$  is adjacent to all  $b_i$ . The labeling  $\chi$  is defined by  $\chi(b_S) = S$  and  $\chi(b_i) = P_i \cup S$ . Hence, the treewidth  $\text{tw}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P}))$  for every YES-instance is bounded above by  $k + c - 1$ . We now use Bodlaender's theorem to check in fpt-time whether  $\text{tw}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})) \leq k + c - 1$ . If the answer is NO, there cannot be a backdoor set of size  $k$ .

Otherwise, we construct an MSO formula  $\varphi$  (whose size depends only on  $c$  and  $k$ ) which checks whether there is a set of vertices  $S \subseteq V$  such that there is no weakly connected component of size larger than  $c$  after deleting the vertices in  $S$ . Then we can check in fpt-time whether  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P})$  satisfies  $\varphi$  by using Courcelle's theorem. It remains to present the MSO formula  $\varphi$ . For this, let  $\text{adj}(x, y)$  be the symmetric adjacency relation stating that vertex  $x$  is adjacent to/from vertex  $y$ . Then,  $\varphi$  is defined as follows:

$$\begin{aligned} \varphi := & \exists S \subseteq V \left[ |S| \leq k \wedge \neg \left( \exists x_1, \dots, x_{c+1} \in V \setminus S \wedge \right. \right. \\ & \left. \left. \bigwedge_{1 \leq i < j \leq c+1} x_i \neq x_j \wedge \bigwedge_{1 \leq i < j \leq c+1} \left( \text{adj}(x_i, x_j) \vee \bigvee_{l \in \{2, \dots, c\}} \psi_l(x_i, x_j) \right) \right) \right] \\ \text{where } \psi_l(x, x') := & \left( \exists y_1, \dots, y_{l-1} \in V \setminus S \left( \text{adj}(x, y_1) \right. \right. \\ & \left. \left. \wedge \text{adj}(y_{l-1}, x') \wedge \bigwedge_{m \in [l-2]} \text{adj}(y_m, y_{m+1}) \right) \right) \end{aligned}$$

It is easy to verify that  $\varphi$  is satisfiable if and only if  $\langle \mathbb{P}, k \rangle$  is a YES-instance.  $\square$

Notice that although the previous result shows fixed-parameter tractability, the direct applicability might be limited, due to the running time caused by the function  $f(k)$ . However, this fpt-result can be seen as the starting point of the development of more sophisticated fpt-algorithms with improved runtime. Such algorithms can be obtained, e.g., by using dynamic programming or other advanced techniques.

In the next result we show that although finding a small backdoor is fpt, the evaluation remains hard for this type of backdoor. This is shown by a reduction from the PARTITIONED CLIQUE problem, which is known to be W[1]-complete (Pietrzak 2003).

PARTITIONED CLIQUE

*Instance:* An integer  $k$ , a  $k$ -partite graph  $G = (V, E)$  with partition  $\{V_1, \dots, V_k\}$  of  $V$  into sets of equal size.

*Parameter:*  $k$ .

*Question:* Does  $G$  have a  $k$ -clique, i.e., a set  $V' \subseteq V$  of  $k$  vertices such that  $\forall u, v \in V'$ , with  $u \neq v$  there is an edge  $\{u, v\} \in E$ , and  $\forall i \in [k]$  it holds that  $|V' \cap V_i| = 1$ ?

**Theorem 4.** *c*-CAUSAL EVALUATION[VARIABLES] and *c*-BOUNDED CAUSAL EVALUATION[VARIABLES] are W[1]-hard even for planning instances with bounded domain.

*Proof.* We reduce from PARTITIONED CLIQUE. Let  $G' = (V, E)$  be a  $k$ -partite graph where  $V$  is partitioned into  $V_1, \dots, V_k$ . We construct an instance  $\langle \mathbb{P}, S \rangle$  of *c*-CAUSAL EVALUATION[VARIABLES] and  $\langle \mathbb{P}, S, k' \rangle$  of *c*-BOUNDED CAUSAL EVALUATION[VARIABLES] in the following way.

Let  $k' := k + \binom{k}{2}$  and  $k'' := k + k'$ . The variables  $V'$  are the union of three kinds of variables: (i) the variables corresponding to the vertices of the graph  $V = \{v_1, \dots, v_n\}$ , (ii) pair-variables  $V_p$  of the form  $p_{i,j}$  for  $1 \leq i < j \leq k$ , and (iii) counter variables  $V_c = \{c_1, \dots, c_{k'}\}$ .

The actions  $A$  are the union of two types of actions: First, the actions  $A_v$  where for each  $v \in V$  and  $l \in [k']$  we introduce an action  $a_v^l$  with  $\text{pre}(a_v^l)[c_l] = 0$  and  $\text{eff}(a_v^l)[c_l] = \text{eff}(a_v^l)[v] = 1$ . This type of actions allows to select  $k$  vertices into the set  $V'$ . Second, the actions  $A_p$  where for each  $l \in [k']$  and each edge  $\{v_i, v_j\} \in E$  such that  $1 \leq i < j \leq n$ ,  $v_i \in V_{i'}$ ,  $v_j \in V_{j'}$ , and  $i', j' \in [k]$  (with  $i' \neq j'$ ), we introduce an action  $a_{i,j}^l$  with  $\text{pre}(a_{i,j}^l)[c_l] = 0$ ,  $\text{pre}(a_{i,j}^l)[v_i] = \text{pre}(a_{i,j}^l)[v_j] = 1$ , and  $\text{eff}(a_{i,j}^l)[c_l] = \text{eff}(a_{i,j}^l)[p_{i',j'}] = 1$ . The intuition of this type of actions is as follows. These actions allow to set a pair variable  $p_{i',j'} \in V_p$  to 1 whenever there is an edge between  $v_i \in V_{i'}$  and  $v_j \in V_{j'}$  in  $G'$  and the variables  $v_i$  and  $v_j$  have been selected previously using actions of the first type.

The initial state is defined as  $I = 0^{n+k''}$ , whereas in the goal  $G$  we set the variables in  $V_p$  to 1 and all others to undefined ( $\mathbf{u}$ ). Now let  $\mathbb{P} := \langle V', \{0, 1\}, A, I, G \rangle$ ,  $S := V_p \cup V_c$ .

It is easy to verify that deleting the variables in  $S$  yields a causal graph where the maximum size of the components is one. Notice that since  $|S| = 2k' - k$ ,  $|S|$  is bounded by a function of  $k$ . Furthermore, it is straightforward to check the equivalence of the following three statements: (i)  $\langle \mathbb{P}, S \rangle$  is a YES-instance (ii)  $\langle \mathbb{P}, S, k'' \rangle$  is a YES-instance (iii)  $\langle k, G', \{V_1, \dots, V_k\} \rangle$  is a YES-instance.  $\square$

Note that the previous construction could be simplified for  $c$ -BOUNDED CAUSAL EVALUATION[VARIABLES] by removing the counter variables.

The hardness results for  $c$ -CAUSAL EVALUATION[VARIABLES] and  $c$ -BOUNDED CAUSAL EVALUATION[VARIABLES] indicate that we should continue and consider further types of backdoors in order to obtain the desirable case where both detection and evaluation are fpt.

### 3.2 Action-Deletion-Backdoors

In this section we show our two main positive results, namely, that SAS<sup>+</sup> PLANNING is fixed-parameter tractable parameterized by the number of actions one needs to delete in order to obtain a causal graph with constant size components for planning instances with bounded domain variables. The same holds true for BOUNDED SAS<sup>+</sup> PLANNING even for planning instances with an unbounded domain. To obtain these results we first show that the detection problem for action-deletion-backdoors is fixed-parameter tractable (Theorem 7). We then show that this also holds true for the evaluation problem for planning instances of bounded domain and for instances of bounded planning in Theorem 9 and 10, respectively.

For the action-deletion-backdoors, i.e., the setting where actions are removed to obtain components of size at most  $c$ , the problems  $c$ -CAUSAL DETECTION[ACTIONS],  $c$ -CAUSAL EVALUATION[ACTIONS], and  $c$ -BOUNDED CAUSAL EVALUATION[ACTIONS] are defined analogously to the respective

problems for variable-deletion-backdoors. Notice that  $S$  now denotes a set of actions.

The property of having small component size in the causal graph is fragile with respect to adding actions: For instance, consider the case where a “reset” action is added to a planning instance with small component size. The “reset” action sets all variables according to their value in the initial state. As a result the causal graph contains only one big component. However, observe that the size of an action-deletion-backdoor for this instance is one.

As in the previous section we start by showing classical hardness of the detection problem. To this end, we make use of the 3-DIMENSIONAL MATCHING problem, which is well-known to be NP-complete (Garey and Johnson 1979).

#### 3-DIMENSIONAL MATCHING

*Instance:* Three disjoint sets  $X, Y, Z$  of the same cardinality, a set of triples  $T \subseteq X \times Y \times Z$ , and a  $k \in \mathbb{N}$ .

*Question:* Is there a set  $T' \subseteq T$  with  $|T'| \geq k$  such that no distinct  $t, t' \in T'$  agree on at least one coordinate?

**Theorem 5.** *For every  $c \geq 3$  the problem  $c$ -CAUSAL DETECTION[ACTIONS] is NP-hard.*

*Proof.* We reduce from 3-DIMENSIONAL MATCHING to  $c$ -CAUSAL DETECTION[ACTIONS] for every  $c \geq 3$ . Let  $\langle (X, Y, Z), T, k \rangle$  be an instance of 3-DIMENSIONAL MATCHING with  $|X \cup Y \cup Z| = n$  and  $|T| = t$  and let  $c \geq 3$ . We construct an instance  $\langle \mathbb{P}, k \rangle$  of  $c$ -CAUSAL DETECTION[ACTIONS] in the following way: We set  $D = \{0, 1\}$ ,  $I = 0^n$ ,  $G = 1^n$ ,  $V = \{v_x \mid x \in (X \cup Y \cup Z)\} \cup \{v_t^i \mid t \in T \text{ and } 3 < i \leq c\}$ . Furthermore, the set  $A$  of actions contains one action  $a_t$  for every triple  $t \in T$  with  $\text{eff}(a_t)[v_x] = 1$  and  $\text{eff}(a_t)[v_t^i] = 1$  for  $x$  in  $t$  and  $3 < i \leq c$ . Notice that the  $v_t^i$  variables are used to pad the size of the components to  $c$  if we reduce to  $c$ -CAUSAL DETECTION[ACTIONS] with  $c > 3$ .

We claim that  $\langle (X, Y, Z), T, k \rangle$  is a YES-instance of 3-DIMENSIONAL MATCHING if and only if the instance  $\langle \mathbb{P}, t - k \rangle$  with  $\mathbb{P} = \langle V, D, A, I, G \rangle$  is a YES-instance of  $c$ -CAUSAL DETECTION[ACTIONS].

Suppose that  $\langle (X, Y, Z), T, k \rangle$  is a YES-instance of 3-DIMENSIONAL MATCHING and let  $T'$  be a set of at least  $k$  triples witnessing this. It is straightforward to check that the set  $A' = \{a_t \mid t \in T \setminus T'\}$  satisfies  $\text{cc-size}(\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus A')) \leq c$  and hence  $A'$  is a solution of  $\langle \mathbb{P}, t - k \rangle$ .

For the reverse direction suppose that  $\langle \mathbb{P}, t - k \rangle$  is a YES-instance of  $c$ -CAUSAL DETECTION[ACTIONS] and let  $A'$  be a set of at most  $t - k$  actions witnessing this. We claim that the set  $T' = \{t \mid a_t \notin A'\}$  satisfies  $t \cap t' = \emptyset$  for every  $t$  and  $t'$  in  $T'$ . Suppose for a contradiction that there are  $t$  and  $t'$  in  $T'$  with  $t \cap t' \neq \emptyset$ . It follows that the graph  $\mathcal{G}_{\text{CAUSAL}}(G \setminus A')$  contains a component that contains all vertices in  $t \cup t'$  and additionally the vertices  $\{v_{t''}^i \mid t'' \in \{t, t'\} \text{ and } 3 < i \leq c\}$ . Since we can assume w.l.o.g. that  $t \neq t'$  and thus  $|t \cup t'| \geq 4$  this contradicts our assumption that all components of the graph  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus A')$  have cardinality at most  $c$ .  $\square$

The following technical lemma is very useful to simplify the proof of the next theorem, an fpt-result for the detection problem.

*c*-BALANCED LABEL SEPARATOR

*Instance:* An undirected edge-labeled multigraph  $G$  (without self-loops), a labeling function  $\lambda : E(G) \rightarrow \mathbb{N}$ , and an integer  $k$ .

*Parameter:*  $k$ .

*Question:* Is there a set  $L$  of at most  $k$  labels such that  $\text{cc-size}(G \setminus \lambda^{-1}(L)) \leq c$ ?

**Lemma 6.** *c*-BALANCED LABEL SEPARATOR is *fpt*.

With help of this lemma, we now show the desired result.

**Theorem 7.** *c*-CAUSAL DETECTION[ACTIONS] is *fpt*.

*Proof.* We reduce *c*-CAUSAL DETECTION[ACTIONS] to *c*-BALANCED LABEL SEPARATOR. Let  $\langle \mathbb{P}, k \rangle$  be an instance of CAUSAL DETECTION[ACTIONS]. We construct an equivalent instance  $\langle G, \lambda, k \rangle$  of *c*-BALANCED LABEL SEPARATOR as follows. Let  $G$  be the undirected edge-labeled multigraph with vertex set  $V$  and an edge  $e$  labeled with  $a$  between two variables  $v$  and  $v'$  for every action  $a \in A$  with  $\text{pre}(a)[v] \neq \mathbf{u}$  and  $\text{eff}(a)[v'] \neq \mathbf{u}$  and for every action  $a \in A$  with  $\text{eff}(a)[v] \neq \mathbf{u}$  and  $\text{eff}(a)[v'] \neq \mathbf{u}$ . It is straightforward to check that  $\langle \mathbb{P}, k \rangle$  is a YES-instance of *c*-CAUSAL DETECTION[ACTIONS] if and only if  $\langle G, \lambda, k \rangle$  is a YES-instance of *c*-BALANCED LABEL SEPARATOR.  $\square$

After this promising result for the detection problem, we turn to the evaluation problem. In case neither the plan length nor the size of the domain of the variables is bounded, evaluation remains hard even for action-deletion-backdoors. However, if either the domain or the plan length is bounded we can show two *fpt*-results for the evaluation problem.

**Theorem 8.** 9-CAUSAL EVALUATION[ACTIONS] is paraNP-hard even if the global actions (the actions in the backdoor  $S$ ) have no preconditions.

*Proof (sketch).* The proof is via a reduction from 3-SAT. Let  $\Phi$  be a 3-CNF formula with variables  $x_1, \dots, x_n$ . We construct an instance  $\langle \mathbb{P}, S \rangle$  of 9-CAUSAL EVALUATION[ACTIONS] where  $S$  contains 5 “global actions”  $s_1, \dots, s_5$  and  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P}) \setminus S$  contains one component for each clause of  $\Phi$  and one “global” component (each component contains at most 9 variables). The main idea is that the global component guesses an assignment  $\alpha$  for the variables of  $\Phi$  and forces a sequence of global actions that represents the guessed assignment. In particular, for every variable  $x_i$ , if  $\alpha(x_i) = 0$ , then the following sequence of global actions is forced by the global component:

$$\underbrace{\langle s_1, s_3, \dots, s_1, s_3 \rangle}_{(i-1)\text{-times } s_1, s_3}, s_4, s_3, \underbrace{\langle s_1, s_3, \dots, s_1, s_3 \rangle}_{(n-i)\text{-times } s_1, s_3}, s_5$$

If  $\alpha(x_i) = 1$ , then the following sequence of global actions is forced by the global component:

$$\underbrace{\langle s_2, s_3, \dots, s_2, s_3 \rangle}_{(i-1)\text{-times } s_2, s_3}, s_4, s_3, \underbrace{\langle s_2, s_3, \dots, s_2, s_3 \rangle}_{(n-i)\text{-times } s_2, s_3}, s_5$$

Observe that the above sequences uniquely identify the variable  $x_i$  and its assignment  $\alpha(x_i)$  for every variable  $x_i$ . The total sequence of global actions forced by the global component is then the concatenation of the above sequences for

every variable  $x_i$ . The components of the clauses now ensure that every clause is satisfied by the assignment chosen by the global component. They do so by allowing only sequences of global actions that correspond to a satisfied literal of the respective clause to lead to their (local) goal state. Due to space limitations we had to omit the details of the construction and a rigorous proof.  $\square$

**Theorem 9.** *c*-CAUSAL EVALUATION[ACTIONS] is *fpt* for planning instances with bounded domain.

*Proof.* Let  $\langle \mathbb{P}, S \rangle$  with  $\mathbb{P} = \langle V, D, A, I, G \rangle$  be an instance of *c*-CAUSAL EVALUATION[ACTIONS] such that  $|D| \leq d$  for some constant  $d$ .

We say that two instances  $\mathbb{P}_1 = \langle V_1, D_1, A_1, I_1, G_1 \rangle$  and  $\mathbb{P}_2 = \langle V_2, D_2, A_2, I_2, G_2 \rangle$  of SAS<sup>+</sup> PLANNING are *isomorphic* if there is a bijection  $f$  from  $V_1 \cup D_1 \cup A_1$  to  $V_2 \cup D_2 \cup A_2$  such that: (i)  $f(v) \in V_2$  for every  $v \in V_1$ ,  $f(d) \in D_2$  for every  $d \in D_1$ , and  $f(a) \in A_2$  for every  $a \in A_1$ , (ii) for every  $a \in A_1$  and  $v \in V_1$  it holds that  $f(\text{pre}(a)[v]) = \text{pre}(f(a))[f(v)]$  and  $f(\text{eff}(a)[v]) = \text{eff}(f(a))[f(v)]$ , and (iii) for every  $v \in V_1$  it holds that  $f(I_1[v]) = I_2[f(v)]$  and  $f(G_1[v]) = G_2[f(v)]$ .

Let  $C_1$  and  $C_2$  be two weakly connected components of the graph  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$ . We say that  $C_1$  and  $C_2$  are *globally equivalent*, denoted by  $C_1 \equiv C_2$ , if there is an isomorphism  $f$  between  $\mathbb{P}[C_1]$  and  $\mathbb{P}[C_2]$  such that  $f(s) = s$  for every  $s \in S$ .

*Claim 1.* The number of equivalence classes of the components of  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$ , with respect to  $\equiv$ , is at most  $c_G = c \cdot ((d+1)^{2c})^{|S|+2} \cdot 2^{(d+1)^{2c}}$ .

The claim follows from the following observations for every component  $C$  of  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$ : (i)  $C$  has at most  $c$  variables, (ii) there are at most  $d^c \cdot (d+1)^c \leq (d+1)^{2c}$  possible configurations for the initial state and for the goal state on the variables of  $C$ , (iii) there are at most  $(d+1)^{2c}$  possible combinations of preconditions and effects for any action in  $S$  on the variables of  $C$ , (iv) there are at most  $(d+1)^{2c}$  possible distinct actions defined only on the variables of  $C$ , and (v) there are at most  $2^{(d+1)^{2c}}$  distinct sets of actions defined only on the variables of  $C$ .

*Claim 2.* Let  $C_1$  and  $C_2$  be two globally equivalent components of  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$ , let  $V_1$  be the variables corresponding to the vertices of  $C_1$ , and let  $\mathbb{P}' = \mathbb{P} \setminus V_1$ . Then,  $\mathbb{P}$  is a YES-instance if and only if so is  $\mathbb{P}'$ .

Suppose that there is a plan  $\omega$  for  $\mathbb{P}$ . Then it is straightforward to verify that the plan  $\omega'$  obtained from  $\omega$  after deleting all occurrences of actions that have at least one precondition or one effect in  $C_1$  and are not in  $S$ , is a plan for  $\mathbb{P}'$ .

For the reverse direction, let  $\omega'$  be a plan for  $\mathbb{P}'$  and let  $f$  be an isomorphism from  $\mathbb{P}[C_2]$  to  $\mathbb{P}[C_1]$  such that  $f(s) = s$  for every  $s \in S$ . Furthermore, let  $\omega$  be obtained from  $\omega'$  after replacing every occurrence of an action  $a$  in  $\mathbb{P}'$  that has at least one precondition or one effect in  $C_2$  and is not in  $S$ , with the sequence  $(a, f(a))$ . Again it is straightforward to check that  $\omega$  is a plan for  $\mathbb{P}$ .

The following algorithm solves the SAS<sup>+</sup> PLANNING problem for  $\mathbb{P}$  in 3 steps.

- 1) Compute the global type, i.e., the equivalence class with respect to  $\equiv$ , of each component  $C$  in  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$ .
- 2) Compute the reduced instance  $\mathbb{P}'$  of SAS<sup>+</sup> PLANNING from  $\mathbb{P}$  by deleting all but one component for each global component type from  $\mathbb{P}$ .
- 3) Compute the state-transition graph of the reduced instance  $\mathbb{P}'$  and solve  $\mathbb{P}'$ . Return YES, if  $\mathbb{P}'$  is a YES-instance and otherwise NO.

The correctness of the algorithm follows immediately from Claim 2.

The running time of the algorithm is obtained as follows. Because of Claim 1, Step 1 needs time  $O(|V| \cdot c_G)$  (recall that  $c_G$  is the number of equivalence classes with respect to  $\equiv$ ), Step 2 can be executed in time  $O(|V|)$  (by using clever data structures), and Step 3 needs time at most  $O(d^{c \cdot c_G})$ , because there are at most  $c \cdot c_G$  variables left in the reduced instance. Hence the total running time of the algorithm is  $O(d^{c \cdot c_G} + |V| \cdot c_G)$ , which shows that  $c$ -CAUSAL EVALUATION[ACTIONS] is fixed-parameter tractable for planning instances with a bounded domain of the variables.  $\square$

**Theorem 10.**  $c$ -BOUNDED CAUSAL EVALUATION[ACTIONS] is *fpt*.

*Proof.* Let  $\langle \mathbb{P}, S, k \rangle$  be an instance of  $c$ -BOUNDED CAUSAL EVALUATION[ACTIONS]. The main idea is to iterate over all  $\binom{k}{l} |S|^l$  sequences of actions from  $S$  that can appear in a plan of length  $l \leq k$  for  $\mathbb{P}$ . For each of those sequences  $\omega$  and for every component  $C$  of  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$  one can then compute the minimal plan (if one exists) for  $\mathbb{P}[C]$  in polynomial time with the help of the state-transition graph of  $\mathbb{P}[C]$  (whose size is at most  $|D|^c$ ). Then  $\mathbb{P}$  has a plan of length at most  $k$  if and only if there is a sequence of actions of  $S$  whose length  $l$  is between 0 and  $k$  such that  $l + s \leq k$ , where  $s$  is the sum of the lengths of the minimal plans for each component of  $\mathcal{G}_{\text{CAUSAL}}(\mathbb{P} \setminus S)$ .  $\square$

Theorems 9 and 10 show in combination with Theorem 7 that the backdoor approach can be used to obtain new *fpt*-algorithms for planning problems. This is because, under action-deletion-backdoors the detection problem as well as three important evaluation problems are efficiently solvable – as long as the size of the backdoor is moderate. The *fpt*-algorithm for planning works as follows. First we search for a small backdoor (detection phase), which is then used in a subsequent step to solve the given planning instance (evaluation phase). Observe that this is the first *fpt*-algorithm for planning with unbounded plan length, which does not depend on the parameters  $|V|$  or  $|A|$ .

## 4 Discussion

In this section we comment briefly on the potential impact to practical applications and the choice of the tractable base class. There has recently been a growing interest in finding tractable fragments of planning using the causal graph (Brafman and Domshlak 2003; Chen and Giménez 2010; Katz and Domshlak 2008; 2010; Giménez and Jonsson 2008; Katz and Keyder 2012)). As pointed out in (Katz and Keyder 2012):

“Such results are not purely of theoretical interest, as the causal graph is used in a variety of practical applications from problem decomposition (Brafman and Domshlak 2006) to the derivation of non-admissible domain-independent heuristics for planning (Helmert 2004).”

We believe that this applies even more to (fixed-parameter) tractable extensions of these fragments obtained by the backdoor approach. In particular, the backdoor approach adds a novel dimension of flexibility to the definition of these tractable classes that allows one to trade efficiency for generality to best suit the particular application. This also allows to use the insights obtained for those tractable classes to solve arbitrary planning instances.

Regarding the choice of the tractable base class, we want to point out that even though the class of instances with bounded component size of the causal graph itself can be seen as rather trivial, the backdoor approach extends its applicability to arbitrary planning instances. Moreover, as shown by (Chen and Giménez 2010) this class is as general as possible for unbounded planning, in the sense that it cannot be properly contained in any other polynomial-time tractable fragment of planning. Nevertheless, considering further fragments remains an interesting direction for future work.

## 5 Conclusion

In this work we have introduced the first two types of backdoor sets for planning. The distance to the tractable fragment was expressed by the number of variables or actions that need to be removed in order to obtain a causal graph of bounded maximum component size. For each backdoor type and each setting of considered SAS<sup>+</sup> planning formalisms (with bounded/unbounded plan length and bounded/unbounded domain of the variables) we have analyzed the (parameterized) complexity of the detection and evaluation problem. In three cases (all settings under action-deletion-backdoors with the exception of unbounded planning with unbounded domain of the variables) we have obtained the most desirable result where detection as well as evaluation are fixed-parameter-tractable. In the remaining cases, we have ruled out the existence of an *fpt*-algorithm by showing hardness for W[1] or paraNP.

We envisage the study of other underlying graph structures (such as the planning graph) to obtain further useful notions of backdoor sets. By employing techniques such as dynamic programming we plan to improve the running time of the *fpt*-result obtained via Courcelle’s theorem. Furthermore, we want to explore additional supporting parameters that help to make the evaluation problem in the variable-deletion setting fixed-parameter tractable.

## Acknowledgments

We want to thank the anonymous reviewers for their constructive comments. Sebastian Ordyniak acknowledges support from the Employment of Newly Graduated Doctors of Science for Scientific Excellence (CZ.1.07/2.3.00/30.0009). Martin Kronegger and Andreas Pfandler were supported by the Austrian Science Fund (FWF): P25518-N23.

## References

- Bäckström, C., and Jonsson, P. 2013. A refined view of causal graphs and component sizes: SP-closed graph classes and beyond. *J. Artif. Intell. Res.* 47:575–611.
- Bäckström, C., and Nebel, B. 1995. Complexity results for SAS<sup>+</sup> planning. *Comput. Intell.* 11:625–656.
- Bäckström, C.; Chen, Y.; Jonsson, P.; Ordyniak, S.; and Szeider, S. 2012. The complexity of planning revisited – a parameterized analysis. In *Proc. AAAI 2012*, 1735–1741. AAAI Press.
- Bodlaender, H. L. 1996. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* 25(6):1305–1317.
- Brafman, R. I., and Domshlak, C. 2003. Structure and complexity in planning with unary operators. *J. Artif. Intell. Res.* 18:315–349.
- Brafman, R. I., and Domshlak, C. 2006. Factored planning: How, when, and when not. In *Proc. AAAI 2006*, 809–814. AAAI Press.
- Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artif. Intell.* 69(1–2):165–204.
- Chen, H., and Giménez, O. 2010. Causal graphs and structurally restricted planning. *J. Comput. Syst. Sci.* 76(7):579–592.
- Courcelle, B. 1990. Graph rewriting: An algebraic and logic approach. In van Leeuwen, J., ed., *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B)*. Elsevier. 193–242.
- Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Springer.
- Downey, R. G.; Fellows, M. R.; and Stege, U. 1999. Parameterized complexity: A framework for systematically confronting computational intractability. In *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future*, volume 49 of *DIMACS Series in Disc. Math. Theor. Comput. Sci.* DIMACS. 49–99.
- Dvorák, W.; Ordyniak, S.; and Szeider, S. 2012. Augmenting tractable fragments of abstract argumentation. *Artif. Intell.* 186:157–173.
- Fichte, J. K., and Szeider, S. 2011a. Backdoors to tractable answer-set programming. In *Proc. IJCAI 2011*, 863–868.
- Fichte, J. K., and Szeider, S. 2011b. Backdoors to tractable answer-set programming. *CoRR* abs/1104.2788.
- Fichte, J. K., and Szeider, S. 2013. Backdoors to normality for disjunctive logic programs. In *Proc. AAAI 2013*, 320–327. AAAI Press.
- Flum, J., and Grohe, M. 2003. Describing parameterized complexity classes. *Inf. Comput.* 187(2):291–319.
- Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*. Springer.
- Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.
- Gaspers, S., and Szeider, S. 2012a. Backdoors to acyclic SAT. In *Proc. ICALP 2012*, volume 7391 of *LNCS*, 363–374. Springer.
- Gaspers, S., and Szeider, S. 2012b. Backdoors to satisfaction. In Bodlaender, H. L.; Downey, R.; Fomin, F. V.; and Marx, D., eds., *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *LNCS*, 287–317. Springer.
- Giménez, O., and Jonsson, A. 2008. The complexity of planning problems with simple causal graphs. *J. Artif. Intell. Res.* 31:319–351.
- Gottlob, G., and Szeider, S. 2008. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *Comput. J.* 51(3):303–325.
- Gregory, P.; Fox, M.; and Long, D. 2008. A new empirical study of weak backdoors. In *Proc. CP 2008*, volume 5202 of *LNCS*, 618–623. Springer.
- Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *Proc. ICAPS 2004*, 161–170. AAAI.
- Katz, M., and Domshlak, C. 2008. New islands of tractability of cost-optimal planning. *J. Artif. Intell. Res.* 32:203–288.
- Katz, M., and Domshlak, C. 2010. Optimal admissible composition of abstraction heuristics. *Artif. Intell.* 174(12–13):767–798.
- Katz, M., and Keyder, E. 2012. Structural patterns beyond forks: Extending the complexity boundaries of classical planning. In *Proc. AAAI 2012*. AAAI Press.
- Kilby, P.; Slaney, J. K.; Thiébaux, S.; and Walsh, T. 2005. Backbones and backdoors in satisfiability. In *Proc. AAAI 2005*, 1368–1373. AAAI Press / The MIT Press.
- Knoblock, C. A. 1994. Automatically generating abstractions for planning. *Artif. Intell.* 68(2):243–302.
- Kronegger, M.; Pfandler, A.; and Pichler, R. 2013. Parameterized complexity of optimal planning: A detailed map. In *Proc. IJCAI 2013*, 954–961. AAAI Press.
- Niedermeier, R. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.
- Nishimura, N.; Ragde, P.; and Szeider, S. 2004. Detecting backdoor sets with respect to Horn and binary clauses. In *Proc. SAT 2004*, 96–103.
- Papadimitriou, C. H. 1994. *Computational complexity*. Addison-Wesley.
- Pfandler, A.; Rümmele, S.; and Szeider, S. 2013. Backdoors to abduction. In *Proc. IJCAI 2013*, 1046–1052. AAAI Press.
- Pietrzak, K. 2003. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. Comput. Syst. Sci.* 67(4):757–771.
- Samer, M., and Szeider, S. 2009. Backdoor sets of quantified Boolean formulas. *J. Autom. Reasoning* 42(1):77–97.
- Williams, R.; Gomes, C.; and Selman, B. 2003. Backdoors to typical case complexity. In *Proc. IJCAI 2003*, 1173–1178. Morgan Kaufmann.