

Variable-Deletion Backdoors to Planning*

Martin Kronegger,¹ Sebastian Ordyniak,² Andreas Pfandler^{1,3}

¹Vienna University of Technology, Vienna, Austria

²Masaryk University, Brno, Czech Republic

³University of Siegen, Siegen, Germany

kronegger@dbai.tuwien.ac.at sordyniak@gmail.com pfandler@dbai.tuwien.ac.at

Abstract

Backdoors are a powerful tool to obtain efficient algorithms for hard problems. Recently, two new notions of backdoors to planning were introduced. However, for one of the new notions (i.e., variable-deletion) only hardness results are known so far. In this work we improve the situation by defining a new type of variable-deletion backdoors based on the extended causal graph of a planning instance. For this notion of backdoors several fixed-parameter tractable algorithms are identified. Furthermore, we explore the capabilities of polynomial time preprocessing, i.e., we check whether there exists a polynomial kernel. Our results also show the close connection between planning and verification problems such as Vector Addition System with States (VASS).

1 Introduction

Classical planning has been an intensively studied field of AI for decades. Until today, planning remains a challenging task due to its high computational complexity. Even propositional STRIPS planning with unbounded plan length is PSPACE-complete. This hardness has led to an intensive study of tractable fragments such as in the work by Bylander (1994) and by Bäckström and Nebel (1995).

The framework of parameterized complexity (Downey and Fellows 1999) has been successfully used to find tractable fragments for problems in AI and beyond. For more details see, e.g., the survey of Gottlob and Szeider (2008). The basic idea of a parameterized complexity analysis is to find parameters capturing the computational hardness in a way such that the combinatorial explosion can be confined to these parameters. Thus, the runtime is measured not only in terms of the input size n but also with respect to one or several parameters k of the instance. The class FPT (*fixed-parameter tractable*) contains all problems that can be solved by an algorithm in time $f(k) \cdot n^{O(1)}$, where n denotes the size of the instance and $f(k)$ is a computable function depending on parameter

k only. Such an fpt-algorithm can be considered efficient as long as the value of the parameter is sufficiently small.

Using these techniques a variety of tractable fragments in form of fpt-algorithms has been revealed for planning (Downey, Fellows, and Stege 1999; Bäckström et al. 2012; Bäckström et al. 2013; Kronegger, Pfandler, and Pichler 2013; Kronegger, Ordyniak, and Pfandler 2014). In the last years the so-called backdoor approach has become a valuable tool to obtain fpt-algorithms, e.g., for SAT (Nishimura, Ragde, and Szeider 2004; Gaspers et al. 2013; Gaspers and Szeider 2013), QBFs (Samer and Szeider 2009), ASP (Fichte and Szeider 2011), Argumentation (Dvorák, Ordyniak, and Szeider 2012), and also for SAS⁺ planning (Kronegger, Ordyniak, and Pfandler 2014). Recently, backdoors were used to construct parameterized reductions for problems that are harder than NP (Fichte and Szeider 2013; Pfandler, Rümmele, and Szeider 2013).

In the backdoor approach, the aim is to find a robust fpt-algorithm that also performs well on instances that are not part of the considered tractable base class, but are sufficiently close to this class. This leads to an algorithm which is applicable to a larger set of instances (when compared to algorithms for the tractable base class). Usually, the performance of such an algorithm scales with the distance to the tractable base class. Solving a problem with the backdoor approach consists of two phases. First, we have to find a backdoor of small size (detection phase), which can afterwards, in the second phase, be used as additional information to guide the decision algorithm for the original problem (evaluation phase). A challenging task is to find suitable notions of backdoors and algorithms building upon these notions such that both phases can be solved efficiently, i.e., by an fpt-algorithm. For planning, it turned out that backdoors cannot be defined in such a direct way as it is possible for SAT and related problems. This is because a suitable notion is needed to describe the underlying structure of the planning instance.

In the work of Kronegger, Ordyniak, and Pfandler (2014) the first two backdoors for planning were introduced. These backdoors measure the number of variables/actions that have to be deleted such that the underlying causal graph decomposes into weakly connected components of small size. Such instances of constant component size are known to be efficiently solvable (Chen and Giménez 2010). For the notion of action-deletion backdoors several fpt-algorithms were ob-

*Supported by the Austrian Science Fund (FWF): P25518-N23, the German Research Foundation (DFG): ER 738/2-1, and by the Employment of Newly Graduated Doctors of Science for Scientific Excellence (CZ.1.07/2.3.00/30.0009).
Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

tained. Among these results is also the first fpt-algorithm for planning with unbounded plan length that limits neither the number of variables or actions in the instance. Unfortunately, for variable-deletion backdoors only parameterized intractability results were shown (W[1]-hardness, to be more precise). Finding fpt-algorithms also for backdoors based on variable-deletion would, however, be very desirable since there are instances for which the notion of deleting variables is stronger than deleting actions.

Main contributions

- We introduce a new type of a variable-deletion backdoors based on the underlying extended causal graph. This new type of backdoor leads to three fpt-algorithms for planning, i.e., fpt-algorithms both for the detection and evaluation problem. We complement these results by showing parameterized intractability for several evaluation problems.
- In our results we show the close connection between planning and verification problems, as we build upon Vector Addition Systems with States (VASS), a formalism strongly related to Petri-nets, to obtain fpt-results for planning.
- We present efficient preprocessing algorithms, polynomial kernels to be more precise, for the detection problems and rule out their existence for the evaluation problems.

2 Preliminaries

We assume the reader to be familiar with the basics of complexity theory and planning.

Parameterized Complexity. In parameterized algorithmics (Downey and Fellows 1999; Flum and Grohe 2006; Niedermeier 2006; Downey and Fellows 2013) the runtime of an algorithm is studied with respect to a parameter $k \in \mathbb{N}$ and input size n . The most favorable class is 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 (fpt). Formally, a *parameterized problem* is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is the input alphabet. Let $L_1 \subseteq \Sigma_1^* \times \mathbb{N}$ and $L_2 \subseteq \Sigma_2^* \times \mathbb{N}$ be parameterized problems. A *parameterized reduction* (or 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 fpt-algorithm w.r.t. parameter k , and (iii) there is a computable function g such that $k' \leq g(k)$, where $(x', k') = P(x, k)$.

The class W[1] captures parameterized intractability and contains all problems that are fpt-reducible to INDEPENDENT SET when parameterized by the size of the solution. The class paraNP (Flum and Grohe 2003) is defined as the class of problems that are solvable by a non-deterministic Turing machine in fpt-time. In our paraNP-hardness proofs, we will make use of the following characterization of paraNP-hardness given by Flum and Grohe (2006), Theorem 2.14: any parameterized problem that remains NP-hard when the parameter is set to some constant is paraNP-hard. The following relations between the parameterized complexity classes hold: $\text{FPT} \subseteq \text{W}[1] \subseteq \text{XP}$, where the class XP contains all problems solvable in time $\mathcal{O}(n^{f(k)})$ for a computable function f . Showing W[1]-hardness for a problem rules out the

existence of an fpt-algorithm under the usual complexity theoretic assumption $\text{FPT} \neq \text{W}[1]$.

Closely related to the search for fpt-algorithms is the search for efficient preprocessing techniques. A *kernelization* algorithm transforms in polynomial time a problem instance (x, k) of a parameterized problem L into an instance (x', k') of L such that (i) $(x, k) \in L$ iff $(x', k') \in L$, (ii) $k' \leq f(k)$, and (iii) the size of x' can be bounded above by $g(k)$, for functions f and g depending only on k . It is easy to show that a parameterized problem is in FPT if and only if there is kernelization algorithm. A *polynomial kernel* is a kernel whose size can be bounded by a polynomial in the parameter.

Planning. Let $V = \{v_1, \dots, v_n\}$ be a finite set of *variables* over a finite *domain* D and let $D^+ = D \cup \{\mathbf{u}\}$, 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 $s[v]$ and the projection of a state s to a set of variables $V' \subseteq V$ is denoted by $s[V']$. A *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 $s \in D^n$. 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. We denote by $V(\mathbb{P})$, $A(\mathbb{P})$ the set V and A , respectively. 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 ℓ). Then ω is a *plan from* s_0 to s_ℓ 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 ω is a *plan for* \mathbb{P} if ω is a plan from I to some goal state. We will study backdoors for the following problems:

Given a SAS⁺ instance \mathbb{P} and a positive integer k the problem BOUNDED SAS⁺ PLANNING is parameterized by k and asks whether \mathbb{P} has a plan of length at most k . The problem SAS⁺ PLANNING is defined analogously without a bound on the plan length k .

Let $\mathbb{P} = \langle V, D, A, I, G \rangle$ be a SAS⁺ instance, $V' \subseteq V$. Then we denote by $\mathbb{P}[V']$ the 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 denote by $\mathbb{P}(V')$ the SAS⁺ instance obtained from $\mathbb{P}[V']$ after deleting all actions that correspond to actions in \mathbb{P} with at least one precondition or effect on a variable in $V \setminus V'$. We write $\mathbb{P} \setminus V'$ for the instance $\mathbb{P}[V \setminus V']$.

Vector Addition System with States. Let $d \in \mathbb{N}$, $u \in \mathbb{Z}^d$, and $i \in \{1, \dots, d\}$. We denote by $u[i]$ the value of the i -th entry (component) of u and by $\|u\|$ the maximum absolute

value of any entry of u , i.e., $\|u\| = \max_{1 \leq j \leq d} |u[j]|$. We also denote by $u + v$, $u - v$, and $u \leq v$ the component-wise summation, subtraction, and comparison of u and v , respectively. A d -VASS \mathbb{V} is a pair $\langle Q, \Delta \rangle$, where Q is a non-empty finite set of *control points* and $\Delta \subseteq Q \times \mathbb{Z}^d \times Q$ is a finite set of *transitions*. A \mathbb{V} -state s is a pair $\langle q, v \rangle$, where $q \in Q$ and $v \in \mathbb{N}^d$. We denote by $\|s\|$ the value $\|v\|$. A *run* in \mathbb{V} is a sequence $\langle \langle q_1, v_1 \rangle, \dots, \langle q_n, v_n \rangle \rangle$ of \mathbb{V} -states such that for every i with $1 \leq i < n$, Δ contains a transition $\langle q_i, v, q_{i+1} \rangle$ with $v_{i+1} = v_i + v$. We say that a \mathbb{V} -state $s_G = \langle q_G, v_G \rangle$ is *coverable* by a \mathbb{V} -state $s_I = \langle q_I, v_I \rangle$ in \mathbb{V} if there is a *covering* of s_G from s_I in \mathbb{V} , i.e., a run $R = \langle \langle q_0, v_0 \rangle, \dots, \langle q_n, v_n \rangle \rangle$ in \mathbb{V} such that $q_0 = q_I$, $v_0 = v_I$, $q_n = q_G$, and $v_n \geq v_G$. We denote by $\|\Delta\|$ the maximum $\|v\|$ over all $v \in \mathbb{Z}^d$ such that $\langle q, v, q' \rangle \in \Delta$ for some $q, q' \in Q$.

Theorem 1 (Bozzelli and Ganty (2011), Theorem 1). *Let $\mathbb{V} = \langle Q, \Delta \rangle$ be a d -VASS and s_I and s_G be two \mathbb{V} -states. If s_G is coverable from s_I in \mathbb{V} , then there is a covering of s_G from s_I in \mathbb{V} of length at most $(|Q| \cdot (\|\Delta\| + \|s_G\| + 2))^{(3d)!+1}$.*

3 A New Type of Backdoors

The backdoor approach is a powerful tool to obtain fpt-algorithms for hard problems. This approach comprises two phases: In the first phase, the *detection phase*, one needs to find a certain set, the *backdoor*, whose size is used to measure the distance of a given instance to a *tractable base class*. After that, in the second phase, which is called the *evaluation phase*, the additional information stored in the backdoor set is used to solve the problem efficiently. The runtime of the evaluation phase usually depends exponentially on the size of the given backdoor. Still, we can consider the algorithm to be efficient as long as the backdoor is reasonably small. Therefore, the size of the backdoor set is usually considered as a parameter. In particular, if detection and evaluation are both fpt, this yields an fpt-algorithm for the whole problem.

Recently, for SAS⁺ PLANNING the first two notions of backdoors were introduced (Kronegger, Ordyniak, and Pfandler 2014), where the so-called *causal graph* (Knoblock 1994; Brafman and Domshlak 2006; Chen and Giménez 2010; Bäckström and Jonsson 2013) is used to capture the structure of the instance. The considered base class contains all instances for which the size of the largest weakly-connected component of the underlying causal graph is bounded by a constant c . For this base class, SAS⁺ PLANNING is solvable in polynomial time (Chen and Giménez 2010).

Kronegger, Ordyniak, and Pfandler (2014) considered two types of backdoors: action-deletion and variable-deletion backdoors. While for both types detection is fpt, the complexity of evaluation varies among these two types. For the former type several fpt-algorithms were found. In contrast, for the latter type only W[1]-hardness results were shown that rule out the existence of an fpt-algorithm under the usual assumptions. Having fpt-algorithms for variable-deletion backdoors would, however, be beneficial as variable-deletion can be stronger than action-deletion. This is because, deleting a single, central variable can have the same effect as deleting a

large set of actions, which can result in a notably smaller backdoor.

The goal of this work is to find an improved type of variable-deletion backdoors, for which detection remains fpt and for which evaluation becomes fpt. To this end, we introduce the *extended causal graph* $\mathcal{G}_E(\mathbb{P})$ of a SAS⁺ instance $\mathbb{P} = \langle V, D, A, I, G \rangle$. $\mathcal{G}_E(\mathbb{P})$ is a directed graph with vertices V that contains an arc (v, v') , for distinct $v, v' \in V$, if there is an action $a \in A$ with (i) $\text{pre}(a)[v] \neq \mathbf{u}$ and $\text{pre}(a)[v'] \neq \mathbf{u}$, or (ii) $\text{pre}(a)[v] \neq \mathbf{u}$ and $\text{eff}(a)[v'] \neq \mathbf{u}$, or (iii) $\text{eff}(a)[v] \neq \mathbf{u}$ and $\text{eff}(a)[v'] \neq \mathbf{u}$. With $\text{cc-size}(G)$ we denote the largest weakly connected component of a digraph G . We remark that the notion of the causal graph of \mathbb{P} , denoted by $\mathcal{G}_C(\mathbb{P})$, is similar to the definition above, with the difference that it does not add an edge between variables appearing in the precondition of an action. Therefore, the extended causal graph is more restrictive and hence the tractable fragment mentioned above immediately carries over to the extended causal graph. Furthermore, as studied by Kronegger, Ordyniak, and Pfandler (2014), this base class can also be used for BOUNDED SAS⁺ PLANNING.

In this work, we consider variable-deletion backdoors on the extended causal graph together with four different settings of planning. We study the SAS⁺ PLANNING and the BOUNDED SAS⁺ PLANNING problem in case of a bounded or unbounded domain of the variables, as well as, novel backdoor specific restrictions, such as, instances that do not contain “mixed actions” (i.e., actions having effects both on the backdoor set and at least one component). In our analysis, we present fpt-algorithms, and hardness results for the classes W[1] and paraNP. If a problem is known to be fpt, the next challenge is to find (or rule out the existence of) a method for efficient preprocessing, i.e., a polynomial kernel. Of course this is the most desirable result, as this yields a versatile polynomial time algorithm that produces a small, equivalent instance, which can in turn be processed by any algorithm for the problem. The results obtained in this work are juxtaposed with known results from the literature (Kronegger, Ordyniak, and Pfandler 2014) in Table 1. Due to space limitations several proof details had to be omitted. The proofs for results marked with * are provided in the supplementary material (Kronegger, Ordyniak, and Pfandler 2015).

3.1 Detection

As a first step, we show that detection of our new type of backdoors remains fpt as it was for the version defined over the causal graph. Actually, we prove a stronger result, namely, that there exists a polynomial kernel for the detection problem. This is good news as this lets hope for very efficient detection algorithms. We consider the following problem:

c -EXTENDED CAUSAL DETECTION[VARIABLES]

Instance: A SAS⁺ 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 $\text{cc-size}(\mathcal{G}_E(\mathbb{P} \setminus S)) \leq c$?

Throughout the paper, we abbreviate c -EXTENDED CAUSAL DETECTION[VARIABLES] with c -DET.

Setting	Domain	Detection		Evaluation	
		(un)bounded plan len.	bounded plan len.	unbounded plan length	
variable-del. in \mathcal{G}_C	(un)bounded	in FPT †	W[1]-hard †	W[1]-hard †	
action-del. in \mathcal{G}_C	bounded	in FPT †	in FPT †	in FPT †	
	unbounded	in FPT †	in FPT †	paraNP-hard †	
variable-deletion in \mathcal{G}_E	bounded	in FPT and pk (Thm. 3)	in FPT (Thm. 8) and npk (Thm. 16)	sog: in FPT (Cor. 13) and npk (Thm. 16) nma: in FPT (Thm. 15) and npk (Thm. 16) in general: in XP (Thm. 10) and npk (Thm. 16)	
	unbounded	in FPT and pk (Thm. 3)	W[1]-hard and in XP (Thm. 5)	paraNP-hard (Thm. 4)	

Table 1: Complexity map. Results marked with † are shown in (Kronegger, Ordyniak, and Pfandler 2014). We use the abbreviation (n)pk for “(no) polynomial kernel”, sog for “additionally parameterized by the size of the goal” and nma for “no mixed actions”.

Before we start with our parameterized analysis we need to show that the detection problem is NP-hard. Inspecting the NP-hardness proof for variable-deletion backdoors based on the causal graph (Kronegger, Ordyniak, and Pfandler 2014, Theorem 2), we observe that the construction does not use preconditions in the actions. Thus, the result carries over to backdoors based on the extended causal graph.

Corollary 2. *c-DET is NP-hard even for planning instances with bounded domain.*

In the next result we show that the detection problem is not only fpt when parameterized by the size of the backdoor but also admits a polynomial kernel.

Theorem 3. *c-DET admits a polynomial kernel of size at most $\mathcal{O}((k + k(k + c)c)^2)$ and hence is fpt.*

Proof. Let $\langle \mathbb{P}, k \rangle$ be an instance of *c-DET* over variables $V = V(\mathbb{P})$. We can construct $\mathcal{G}_E(\mathbb{P})$ in polynomial time.

W.l.o.g. we can assume that $\mathcal{G}_E(\mathbb{P})$ is weakly connected, otherwise a deletion set for $\mathcal{G}_E(\mathbb{P})$ can be obtained as the union of deletion sets for every component of $\mathcal{G}_E(\mathbb{P})$. Furthermore, we can assume that the degree of every vertex of $\mathcal{G}_E(\mathbb{P})$ is at most $k + c$. Assume that this is not the case, i.e., there is a vertex v of $\mathcal{G}_E(\mathbb{P})$ with more than $k + c$ neighbors. Then v has to be contained in the deletion set, because otherwise the deletion set would have to contain more than k of its neighbors. Consequently, $\langle \mathbb{P} \setminus \{v\}, k - 1 \rangle$ is equivalent to $\langle \mathbb{P}, k \rangle$. We now show that if $\langle \mathbb{P}, k \rangle$ is a YES-instance, then \mathbb{P} can have at most $k + k(k + c)c$ variables. So assume that $\langle \mathbb{P}, k \rangle$ is a YES-instance and let S be a deletion set of size at most k witnessing this. Because every vertex in S has at most $k + c$ neighbors and $\mathcal{G}_E(\mathbb{P})$ is connected, we obtain that $\mathcal{G}_E(\mathbb{P}) \setminus S$ has at most $k(k + c)$ components and hence $\langle \mathbb{P}, k \rangle$ has at most $k + k(k + c)c$ variables. Thus, we can transform $\langle \mathbb{P}, k \rangle$ in polynomial time into an equivalent instance $\langle \mathbb{P}', k' \rangle$ that contains at most $k'' = k + k(k + c)c$ variables. The following claim shows that also the number of actions can be bounded and thus we obtain a polynomial kernel.

Claim 1 ().* There is an equivalent instance $\langle \mathbb{P}'', k'' \rangle$ with at most k'' variables and at most $(k'')^2$ actions. \square

3.2 Evaluation

After we have shown that detection is fpt, we turn to the evaluation problem. Finding an fpt-result for these problems

would hence give an fpt-result for the whole backdoor approach. This is in particular interesting as no fpt-results for variable-deletion backdoors are known so far (cf. Table 1). The evaluation problems are defined as follows:

<i>c</i> -BOUNDED EXTENDED CAUSAL EVALUATION[VARIABLES]
<i>Instance:</i> A SAS ⁺ instance \mathbb{P} , a positive integer k , and a set S of variables of \mathbb{P} such that $\text{cc-size}(\mathcal{G}_E(\mathbb{P} \setminus S)) \leq c$.
<i>Parameter:</i> $ S + k$
<i>Question:</i> Does \mathbb{P} have a plan of length at most k ?

The problem *c-EXTENDED CAUSAL EVALUATION[VARIABLES]* is defined analogously without a bound on the plan length k . We abbreviate *c-EXTENDED CAUSAL EVALUATION[VARIABLES]*, resp. *c-BOUNDED EXTENDED CAUSAL EVALUATION[VARIABLES]*, with *c-EVAL*, resp. *c-BOUNDED EVAL*.

If neither the plan length nor the domain of the variables is bounded, evaluation is paraNP-hard. However, if only the length is bounded we obtain W[1]-hardness and membership in XP. This yields a polynomial algorithm if the parameter values are fixed to a constant.

Theorem 4 (*). *2-EVAL is paraNP-hard.*

Theorem 5 (*). *c-BOUNDED EVAL is W[1]-hard (even if additionally parameterized by the number of variables) and in XP.*

Proof (sketch). Hardness can be shown by a reduction from PARTITIONED CLIQUE parameterized by the number of partitions, which is known to be W[1]-complete (Pietrzak 2003). The reduction is similar to the one given in (Kronegger, Ordyniak, and Pfandler 2014, Theorem 4) and is therefore omitted. For XP-membership notice that all possible plans of length k can be trivially bounded by $\mathcal{O}(n^k)$, where n is the number of actions in the SAS⁺ instance. \square

The above Theorem shows that *c-BOUNDED EVAL* is contained in XP. In the following we will show that if we consider only *c-BOUNDED EVAL* instances of bounded domain then this can be improved to an fpt-result. Before showing this, we need additional definitions.

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⁺ PLANNING are *isomorphic* if there is a bijection φ from $V_1 \cup D_1 \cup A_1$ to $V_2 \cup D_2 \cup A_2$

such that (i) $\varphi(v) \in V_2$ for every $v \in V_1$, $\varphi(d) \in D_2$ for every $d \in D_1$, and $\varphi(a) \in A_2$ for every $a \in A_1$, (ii) for every $a \in A_1$ and $v \in V_1$ it holds that $\varphi(\text{pre}(a)[v]) = \text{pre}(\varphi(a))[\varphi(v)]$ and $\varphi(\text{eff}(a)[v]) = \text{eff}(\varphi(a))[\varphi(v)]$, and (iii) for every $v \in V_1$ it holds that $\varphi(I_1[v]) = I_2[\varphi(v)]$ and $\varphi(G_1[v]) = G_2[\varphi(v)]$.

In the following let $\mathbb{P} = \langle V, D, A, I, G \rangle$ be a SAS⁺ PLANNING instance and S be a c -Extended Causal Backdoor of \mathbb{P} . We denote by $C(\mathbb{P}, S)$ the set of components of $\mathcal{G}_E(\mathbb{P} \setminus S)$. We say that two components C_1 and C_2 in $C(\mathbb{P}, S)$ are *equivalent*, denoted by $C_1 \equiv C_2$, if there is an isomorphism φ between $\mathbb{P}(V(C_1) \cup S)$ and $\mathbb{P}(V(C_2) \cup S)$ such that $\varphi(s) = s$ for every $s \in S$.

Let $p \in \mathbb{N}$. We denote by $E(\mathbb{P}, S, p)$ the SAS⁺ PLANNING instance obtained from \mathbb{P} by removing any component in excess of p from $C(\mathbb{P}, S)$ for every equivalence class with respect to \equiv . Furthermore, we write $E(\mathbb{P}, S)$ to denote $E(\mathbb{P}, S, 1)$. By “removing a component $C \in C(\mathbb{P}, S)$ from \mathbb{P} ” we mean that all variables in $V(\mathbb{P})$ together with all actions that have at least one precondition or effect in C are removed from \mathbb{P} . Observe that $E(\mathbb{P}, S, p)$ is unique up to isomorphism.

We denote by $ET(\mathbb{P}, S)$ an arbitrary but fixed ordering of the components of $C(E(\mathbb{P}, S), S)$ and by $EV(\mathbb{P}, S)$ the vector $v \in \mathbb{N}^{|ET(\mathbb{P}, S)|}$ such that for every i with $1 \leq i \leq |ET(\mathbb{P}, S)|$, we have that $v[i]$ is equal to the total number of components of $C(\mathbb{P}, S)$ that are equivalent to the component $ET(\mathbb{P}, S)[i]$. Observe that a planning instance \mathbb{P}' with c -Extended Causal Backdoor S' is equivalent to \mathbb{P} if $E(\mathbb{P}, S)$ is isomorphic to $E(\mathbb{P}', S')$ and $EV(\mathbb{P}, S) = EV(\mathbb{P}', S')$. The following lemma gives an upper bound on the number of equivalence classes of the components.

Lemma 6 (*). $|ET(\mathbb{P}, S)|$ is at most $c \cdot (|D| + 1)^{2(c+|S|)}$. $2^{(|D|+1)^{2(c+|S|)}}$.

Our fpt-algorithms use the following general approach: We first compute a reduced instance \mathbb{P}' of the form $E(\mathbb{P}, S, l)$ for some $l \in \mathbb{N}$ that is equivalent to \mathbb{P} in fpt time. We then use the following lemma to show that \mathbb{P}' and hence also \mathbb{P} can be solved in fpt-time. Observe that instead of solving \mathbb{P}' using brute-force (as it is done in the following lemma) one can use state-of-the-art planners or even heuristic approaches to solve \mathbb{P}' . This provides a high degree of freedom for the design of efficient algorithms that use our backdoor set approach.

Lemma 7 (*). Given a SAS⁺ PLANNING instance \mathbb{P} , a c -Extended Causal Backdoor S of \mathbb{P} , and $l \in \mathbb{N}$, we can solve c -EVAL and c -BOUNDED EVAL of $E(\mathbb{P}, S, l)$ in time $\mathcal{O}(|D|^{V(E(\mathbb{P}, S, l))} + |V| \cdot |ET(\mathbb{P}, S)|)$.

With the help of the above two lemmas it is now relatively straightforward to prove the following theorem.

Theorem 8 (*). c -BOUNDED EVAL is fpt for instances with bounded domain.

Given that c -BOUNDED EVAL is fpt for instances with bounded domain, it becomes natural to ask whether the same holds for c -EVAL instances of bounded domain. Even though, we are not able to answer this question in its full generality, we obtain fpt-algorithms for two natural restrictions of this problem, i.e., (i) if we consider the size of the goal as an

additional parameter, and (ii) for instances, which do not contain any mixed actions, i.e., actions that have effects both on S and on some component in $C(\mathbb{P}, S)$. Before showing these two results we need some preparation. In particular, we first show that c -EVAL is in XP for instances of bounded domain.

We denote by $ST(\mathbb{P})$ an arbitrary but fixed ordering of all possible (total) states of \mathbb{P} . Let s be a (total) state of \mathbb{P} and assume that $ET(\mathbb{P}, S) = \langle C_1, \dots, C_{|ET(\mathbb{P}, S)|} \rangle$. Below we define the vector $ESV(\mathbb{P}, S, s)$, which will be important for the following constructions. The vector contains one entry for every type of a component and for every possible state of a component, whose value is equal to the number of components of the corresponding type being in that state (in the global state s). We denote by $ESV-D(\mathbb{P}, S)$ the number $\sum_{i=1}^{|ET(\mathbb{P}, S)|} |ST(\mathbb{P}(V(C_i)))|$. Let i with $1 \leq i \leq |ET(\mathbb{P}, S)|$ and $z \in ST(\mathbb{P}(V(C_i)))$. We denote by $ESV-I(\mathbb{P}, S, C_i, z)$ the number $\sum_{j=1}^{j < i} |ST(\mathbb{P}(V(C_j)))| + j_z$, where $z = ST(\mathbb{P}(V(C_i)))[j_z]$. We denote by $ESV(\mathbb{P}, S, s)$ the vector $v \in \mathbb{N}^{ESV-D(\mathbb{P}, S)}$ such that for every i with $1 \leq i \leq |ET(\mathbb{P}, S)|$ and every $z \in ST(\mathbb{P}(V(C_i)))$, we have that $v[ESV-I(\mathbb{P}, S, C_i, z)]$ is equal to the total number of components C in $C(\mathbb{P}, S)$ that are equivalent to C_i and $z = s[V(C)]$.

We say that two (total) states of the variables of \mathbb{P} , s_1 and s_2 , are *equivalent*, denoted by $s_1 \equiv_S s_2$, if $ESV(\mathbb{P}, S, s_1) = ESV(\mathbb{P}, S, s_2)$ and $s_1[S] = s_2[S]$.

Lemma 9 (*). Let \mathbb{P} be a SAS⁺ PLANNING instance and s_1, s_2 be two equivalent (total) states of the variables of \mathbb{P} . Then, for any plan from s_1 to G in \mathbb{P} , there is a plan from s_2 to G in \mathbb{P} of the same length, and vice versa.

It follows from the above lemma that in order to decide whether \mathbb{P} has a plan (of a certain length) one does not need to distinguish between equivalent states, i.e., one can build a compressed version of the state-transition graph of \mathbb{P} that only contains exactly one representative state for every equivalence class. Furthermore, since the equivalence class of any (total) state s of \mathbb{P} can be represented by the vector $v = ESV(\mathbb{P}, S, s)$ plus the partial state $p = s[S]$, the compressed state-transition graph contains at most as many states as the number of possible vectors v times the number of partial states p . This leads to the following theorem.

Theorem 10 (*). c -EVAL is in XP for planning instances with bounded domain.

Next, we present our fpt-results for c -EVAL for instances with bounded domain. We start by giving an fpt-algorithm for the case where the goal is only defined on the backdoor set S (but not on any component). This result is shown by reducing the c -EVAL problem to the coverability problem for d -VASS. Then we use a recently obtained result that bounds the length of runs in a d -VASS (Bozzelli and Ganty 2011) to solve the aforementioned special case. We want to note that the reduction is interesting on its own as it applies to c -EVAL in its full generality and shows a deep connection between planning and the d -VASS coverability problem, which is usually merely studied in the context of verification.

Given \mathbb{P} and S , we construct a d -VASS $\mathbb{V}(\mathbb{P}, S) = \langle Q, \Delta \rangle$, where $d = ESV-D(\mathbb{P}, S) + 1$, and two $\mathbb{V}(\mathbb{P}, S)$ -states s_I and

s_G such that \mathbb{P} has a plan if and only if s_G is coverable from s_I in $\mathbb{V}(\mathbb{P}, S)$. In the following we use the abbreviation $\mathbb{P}_{\equiv} = \langle V_{\equiv}, D_{\equiv}, A_{\equiv}, I_{\equiv}, G_{\equiv} \rangle$ to denote the instance $\mathbb{E}(\mathbb{P}, S)$. For $q \in \mathbb{N}$ and $v \in \mathbb{N}^q$, we denote by $v^* \in \mathbb{N}^{q+1}$ the vector with $v^*[i] = v[i]$ for every $1 \leq i \leq q$ and $v^*[q+1] = 0$.

Let s be a (partial) state of \mathbb{P} and $V' \subseteq V$. Then, we denote by $Z(\mathbb{P}, s, V')$ an arbitrary but fixed ordering of the (total) states of $\mathbb{P}(V')$ that are compatible with $s[V']$. The d -VASS $\mathbb{V}(\mathbb{P}, S)$ is constructed as follows.

S1 For every state of the variables in S , we add a control point q_s to Q .

S2 For every action $a \in A(\mathbb{P}_{\equiv})$, we distinguish two cases:

- If a has no precondition and no effect on any component in $C(\mathbb{P}_{\equiv}, S)$, then for every $p_S \in Z(\mathbb{P}_{\equiv}, \text{pre}(a), S)$, we add the new control point $q_{p_S, a}$ and we add the transitions $\langle q_{p_S}, \mathbf{0}, q_{p_S, a} \rangle$ and $\langle q_{p_S, a}, \mathbf{0}, q_{e_S} \rangle$ to Δ , where e_S is the result of a in p_S on $\mathbb{P}_{\equiv}(S)$.
- otherwise, i.e., if a has at least one precondition or effect on a component C in $C(\mathbb{P}_{\equiv}, S)$. Then, for every $p_S \in Z(\mathbb{P}_{\equiv}, \text{pre}(a), S)$ and every $p_C \in Z(\mathbb{P}_{\equiv}, \text{pre}(a), V(C))$, let e_S be the result of a in p_S on $\mathbb{P}_{\equiv}(S)$ and let e_C be the result of a in p_C on $\mathbb{P}_{\equiv}(V(C))$. Then we add a new control point $q_{p_S, a}$ and the transitions $\langle q_{p_S}, v_1, q_{p_S, a} \rangle$ and $\langle q_{p_S, a}, v_2, q_{e_S} \rangle$ to Δ , where $v_1 \in \mathbb{N}^d$ is -1 at the $\text{ESV-I}(\mathbb{P}_{\equiv}, S, C, p_C)$ -th entry and 0 on all other entries and $v_2 \in \mathbb{N}^d$ is $+1$ at the $\text{ESV-I}(\mathbb{P}_{\equiv}, S, C, e_C)$ -th entry and 0 on all other entries.

S3 We add a “goal control point” q_G to Q , as well as, a transition $\langle q_s, \mathbf{0}, q_G \rangle$ for every state $s \in Z(\mathbb{P}_{\equiv}, G_{\equiv}, S)$.

S4 Let $\text{ET}(\mathbb{P}_{\equiv}, S) = \{C_1, \dots, C_{|\text{ET}(\mathbb{P}, S)|}\}$. For every i with $1 \leq i \leq |\text{ET}(\mathbb{P}_{\equiv}, S)|$ such that $G_{\equiv}[V(C_i)] \neq \mathbf{u}$, and every $g \in Z(\mathbb{P}_{\equiv}, G_{\equiv}, V(C_i))$, we add the transition $\langle q_G, v, q_G \rangle$, where v is the vector with $v[\text{ESV-I}(\mathbb{P}_{\equiv}, S, C_i, g)] = -1$, $v[d] = +1$, and all other entries are 0 .

S5 We set $s_I = \langle q_{I[S]}, \text{ESV}(\mathbb{P}, S, I)^* \rangle$ and $s_G = \langle q_G, v_G \rangle$, where $v_G[d]$ is equal to the number of components C in $C(\mathbb{P}, S)$ with $G[V(C)] \neq \mathbf{u}$ and all other entries of v_G are set to 0 .

Theorem 11. *Let \mathbb{P} be a SAS⁺ PLANNING instance with goal G , S be a c -Extended Causal Backdoor of \mathbb{P} , $d = \text{ESV-D}(\mathbb{P}, S) + 1$, and $\mathbb{V}(\mathbb{P}, S) = \langle Q, \Delta \rangle$ be the d -VASS obtained from \mathbb{P} and S . Furthermore, let s_I and s_G be two $\mathbb{V}(\mathbb{P}, S)$ -states (as defined above) and g be the number of components C in $C(\mathbb{P}, S)$ with $G[V(C)] \neq \mathbf{u}$. Then, s_I and s_G satisfy the following properties: (P1) \mathbb{P} has a plan of length at most k if and only if there is a covering of s_G from s_I in $\mathbb{V}(\mathbb{P}, S)$ of length at most $2k + g + 2$, (P2) $|Q| \leq |\text{ST}(\mathbb{P}(S))| \cdot (1 + |A(\mathbb{P}_{\equiv})|) + 1$, (P3) $|\Delta| \leq 1$, (P4) $\|s_G\| = g$, and (P5) $\mathbb{V}(\mathbb{P}, S)$ can be constructed in time $\mathcal{O}(|\text{ET}(\mathbb{P}, S)| \cdot (|V| + (|D| + 1)^{2(|S|+e)}))$.*

Proof (sketch). Due to space restrictions we only sketch the proof of (P1). To this end, we need the following claim.

Claim 2 ().* There is a run of length l from s_I to some $\mathbb{V}(\mathbb{P}, S)$ -state $s_e \in \{ \langle q_s, v \rangle \mid s \in \text{ST}(\mathbb{P}(S)) \text{ and } v \in \mathbb{N}^d \}$ if and only if there is a total state s of \mathbb{P} and a sequence ω of

actions of length $\frac{l-1}{2}$ such that s is the result of ω in I and $s_e = \langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^* \rangle$.

Assume that $\omega = \langle a_1, \dots, a_k \rangle$ is a plan for \mathbb{P} that results in the state s . It follows from Claim 2 that there is a run r from s_I to $\langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^* \rangle$ of length $2k + 1$. Because s is a goal state, we obtain from S3 that there is a transition $t = \langle q_{s[S]}, \mathbf{0}, q_G \rangle$. Furthermore, again because s is a goal state, it follows that the sum of all entries of $\text{ESV}(\mathbb{P}, S, s)$ that correspond to a goal state of some component is equal to g . Hence, we obtain a covering of s_G from s_I of length $2k + 1 + 1 + g = 2k + g + 2$, by appending t followed by all the g applicable transitions of the form $\langle q_G, v, q_G \rangle$ (obtained from S4) to r .

For the reverse direction assume that r is a covering of length l of s_G from s_I in $\mathbb{V}(\mathbb{P}, S)$. Since all outgoing transitions from q_G are self-loops, we obtain that $\mathbb{V}(\mathbb{P}, S)$ -states involving q_G only appear (together) at the end of r . Let s_e be the last state reached by r before the first occurrence of a state involving q_G and let l' be the length of the subsequence of r from s_I to s_e . Because the only transitions leading to a state involving q_G are from states in $\{ \langle q_s, v \rangle \mid s \in \text{ST}(\mathbb{P}(S)) \text{ and } v \in \mathbb{N}^d \}$, we obtain that $s_e \in \{ \langle q_s, v \rangle \mid s \in \text{ST}(\mathbb{P}(S)) \text{ and } v \in \mathbb{N}^d \}$ (see S3). It follows from Claim 2 that there is a total state s of \mathbb{P} and a sequence of actions ω of length $\frac{l'-1}{2}$ such that s is the result of ω in I and $s_e = \langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^* \rangle$. We claim that ω is a plan of length $\frac{l-g-2}{2}$ for \mathbb{P} . Since the only transitions leading to q_G are of the form $\langle q_{s'}, \mathbf{0}, q_G \rangle$ where $s' \in Z(\mathbb{P}_{\equiv}, G_{\equiv}, S)$ (see S3), we obtain that $s[S]$ is compatible with G . Let $\langle q_G, v \rangle$ be the last $\mathbb{V}(\mathbb{P}, S)$ -state visited by r . Because r covers s_G , we obtain that $v[d] \geq g$. Because all transitions that increase the value of the d -th dimension in $\mathbb{V}(\mathbb{P}, S)$ also decrease the value of a dimension that corresponds to some goal state of some component by the same amount (see S4), it follows that the number of components, whose state in s is compatible with the goal state, is at least g . Because g is equal to the total number of components in $C(\mathbb{P}, S)$ for which the goal state is defined (see S5), we obtain that all of these components have reached a goal state in s . Hence, s is compatible with G . Because exactly $g + 1$ transitions are executed by r after the state s_e , we obtain that $l' = l - 1 - g$ and hence the length of ω is $\frac{l'-1}{2} = \frac{l-g-2}{2}$, as required. \square

Theorem 12. *c -EVAL is fpt for instances with bounded domain and no goals in the components.*

Proof. Let $\mathbb{I} = \langle \mathbb{P}, S \rangle$ be an instance of c -EVAL. Furthermore, let $\mathbb{V}(\mathbb{P}, S)$ be the d -VASS and s_I and s_G be the two $\mathbb{V}(\mathbb{P}, S)$ -states as defined above, where $d = \text{ESV-D}(\mathbb{P}, S) + 1$. Because of Proposition 1, we obtain that if s_G is coverable from s_I in $\mathbb{V}(\mathbb{P}, S)$, then there is a covering of s_G from s_I of length at most $l = (|Q| \cdot (|\Delta| + \|s_G\| + 2))^{(3d)+1}$. Using our assumption that G is not defined on any component in $C(\mathbb{P}, S)$ together with Theorem 11 (P1), we obtain that any minimal plan for \mathbb{P} has length at most $\frac{l-2}{2}$.

Because every action in a plan for \mathbb{P} can change the state of at most one component at a time and we assume that there

are no goals defined on the components of $C(\mathbb{P}, S)$, we obtain that \mathbb{P} is equivalent to the instance $\mathbb{P}' = E(\mathbb{P}, S, \frac{l-2}{2})$. It follows from Lemma 7 that we can solve \mathbb{P} in time $\mathcal{O}(|D|^{V(\mathbb{P}')} + |V| \cdot |\text{ET}(\mathbb{P}, S)|)$. To see that this is an fpt-algorithm it only remains to show that $V(E(\mathbb{P}, S, \frac{l-2}{2}))$ can be bounded by a function of c , $|D|$, and $|S|$. In particular, because $|V(E(\mathbb{P}, S, \frac{l-2}{2}))| \leq |S| + c \cdot |\text{ET}(\mathbb{P}, S)| \cdot \frac{l-2}{2}$, it is sufficient to bound $|\text{ET}(\mathbb{P}, S)|$ and l by a function of c , $|D|$, and $|S|$. By Lemma 6 this clearly holds for $|\text{ET}(\mathbb{P}, S)|$. Furthermore, using our assumption that G is not defined on any component in $C(\mathbb{P}, S)$ together with Theorem 11 (P2, P3, and P4), we obtain that $l \leq (3 \cdot |Q|)^{(3d)^{l+1}} \leq (3 \cdot (|\text{ST}(\mathbb{P}(S))| \cdot (|A(\mathbb{P}_{\equiv})| + 1) + 1))^{(3d)^{l+1}}$, where $|A(\mathbb{P}_{\equiv})| \leq (|D| + 1)^{2(c+|S|)} |\text{ET}(\mathbb{P}, S)|$, $|\text{ST}(\mathbb{P}(S))| \leq |D|^{|S|}$, and $d = \text{ESV-D}(\mathbb{P}, S) + 1 \leq |\text{ET}(\mathbb{P}, S)| \cdot |D|^c$, as required. \square

Corollary 13. *c -EVAL is fpt for bounded domain and when additionally parameterized by the size of the goal.*

With help of the next lemma we show that c -EVAL is fpt for bounded domain if there are no mixed actions.

Lemma 14 (*). *Let $\mathbb{I} = \langle \mathbb{P}, S \rangle$ be an instance of c -EVAL that contains no mixed actions. Then \mathbb{I} is equivalent to the c -EVAL instance $\mathbb{I}' = \langle \mathbb{P}', S \rangle$, where $\mathbb{P}' = E(\mathbb{P}, S, m)$ and $m = \max_{T \in \text{ET}(\mathbb{P}, S)} |\text{ST}(\mathbb{P}(V(T)))|$.*

Proof (sketch). For the forward direction suppose that ω is a plan for \mathbb{P} . The main idea to obtain a plan for \mathbb{P}' is to make efficient use of the limited amount of components available and to ensure at the same time that every state of a component that will be used at a later stage of the plan remains accessible. Informally, this is achieved by avoiding to keep multiple components that are in the same state for as long as possible. In this way we are able to construct a valid sequence of actions ω'' for \mathbb{P}' whose application in I' results in a state s'' such that (1) $s''[S]$ is compatible with $G'[S]$, (2) for every component C in $C(\mathbb{P}', S)$, either $s''[V(C)] = I'[V(C)]$ or $s''[V(C)]$ is compatible with $G'[V(C)]$, and (3) for every $T \in \text{ET}(\mathbb{P}', S)$ there is a component $C \in C(\mathbb{P}', S)$ equivalent to T such that $s''[V(C)]$ is compatible with $G'[V(C)]$. The details for the construction of ω'' can be found in (Kronegger, Ordyniak, and Pfandler 2015). Note that ω'' almost constitutes a plan for \mathbb{P}' . It only remains to show how to extend ω'' to ensure that also the components that are still in the initial state after applying ω'' to I' can reach a goal state. For every $C \in C(\mathbb{P}', S)$ with $s''[V(C)] = I'[V(C)]$, we denote by C' a component in $C(\mathbb{P}', S)$ equivalent to C such that $s''[V(C')]$ is compatible with the goal state (which must exist due to (3)). Then, we obtain a plan for \mathbb{P}' by applying the following procedure for every component $C \in C(\mathbb{P}', S)$ with $s''[V(C)] = I'[V(C)]$: For every occurrence of an action a' in ω'' that has at least one effect on C' , we insert (after the occurrence of a') the unique action a in $\mathbb{P}'(S \cup V(C))$ that corresponds to a' . This concludes the proof of the forward direction.

For the reverse direction suppose that ω' is a plan for \mathbb{P}' . For every $C \in C(\mathbb{P}, S) \setminus C(\mathbb{P}', S)$, we denote by C'' one of the components in $C(\mathbb{P}', S)$ that is equivalent to C . Then, we obtain a plan for \mathbb{P} by applying the following procedure

for every component $C \in C(\mathbb{P}, S) \setminus C(\mathbb{P}', S)$: For every occurrence of an action a' in ω' that has at least one effect on C'' , we insert (after the occurrence of a') the unique action a in $\mathbb{P}(S \cup V(C))$ that corresponds to a' . \square

Theorem 15. *c -EVAL is fpt for instances with bounded domain and no mixed actions.*

Proof. Let $\mathbb{I} = \langle \mathbb{P}, S \rangle$ be an instance of c -EVAL, and $m = \max_{T \in \text{ET}(\mathbb{P}, S)} |\text{ST}(\mathbb{P}(V(T)))|$. Because of Lemma 14, it holds that \mathbb{I} is equivalent to the instance $\mathbb{I}' = \langle \mathbb{P}', S \rangle$, where $\mathbb{P}' = E(\mathbb{P}, S, m)$. It follows from Lemma 7 that we can solve \mathbb{P} in time $\mathcal{O}(|D|^{V(\mathbb{P}')} + |V| \cdot |\text{ET}(\mathbb{P}, S)|)$. Because \mathbb{P}' has at most $|S| + c \cdot |\text{ET}(\mathbb{P}, S)| \cdot m$ variables, it now follows from the upper bound on $|\text{ET}(\mathbb{P}, S)|$ given in Lemma 6 and the fact that $m \leq |D|^c$ that c -EVAL is fpt. \square

In contrast to the c -DET problem, for which we showed a polynomial kernel result, the next result shows that it is very unlikely that there exists a polynomial kernel for the evaluation problems c -EVAL and c -BOUNDED EVAL.

Theorem 16 (*). *Neither c -EVAL nor c -BOUNDED EVAL with bounded domain admit a polynomial kernel unless $\text{coNP} \subseteq \text{NP/poly}$. This even holds for c -EVAL without goals on components and mixed actions.*

We want to highlight that the fpt-results of Theorem 12, Corollary 13 and Theorem 15 do not hold if the causal graph is considered instead of the extended causal graph. Observe that for c -EVAL with bounded domain and without goals in the components (Theorem 12) or additionally parameterized by the size of the goal (Corollary 13), $\text{W}[1]$ -hardness is already shown in the construction of the proof of Theorem 4 in the work of Kronegger, Ordyniak, and Pfandler (2014). By small modifications of this construction one can show $\text{W}[1]$ -hardness for c -EVAL with bounded domain but without mixed actions (Theorem 15). Thus, our notion of backdoors yields indeed an improvement when compared to the notion of variable-deletion backdoors considered previously (Kronegger, Ordyniak, and Pfandler 2014).

4 Conclusion

We have introduced a new type of variable-deletion backdoors and have shown three kinds of results for the detection and evaluation problem in various settings: In three cases we were able to present (i) fpt-algorithms, which are complemented by (ii) parameterized intractability results. Furthermore, for the problems that are fpt, we have answered the question (iii) whether efficient preprocessing is possible, i.e., whether there is a polynomial kernel. With help of this new type of backdoor, we could obtain fpt-results for problems, which remain $\text{W}[1]$ -hard under the notion based on the causal graph. This answers, in particular, one of the open questions asked in (Kronegger, Ordyniak, and Pfandler 2014).

Since the connection to verification allowed to build upon results for the VASS formalism, it would be interesting to closer investigate this connection. This might also help to solve the open problem whether c -EVAL with bounded domain of the variables is fpt. Solving this open problem would also provide novel insights into the coverability problem for

VASS. Finally, studying new types of backdoors for planning is also an interesting direction for further research.

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⁺ 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. AAI 2012*, 1735–1741. AAAI Press.
- Bäckström, C.; Jonsson, P.; Ordyniak, S.; and Szeider, S. 2013. Parameterized complexity and kernel bounds for hard planning problems. In *Proc. CIAC 2013*, volume 7878 of LNCS, 13–24. Springer.
- Bozzelli, L., and Ganty, P. 2011. Complexity analysis of the backward coverability algorithm for VASS. In *Proc. RP 2011*, volume 6945 of LNCS, 96–109. Springer.
- Brafman, R. I., and Domshlak, C. 2006. Factored planning: How, when, and when not. In *Proc. AAI 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.
- Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Springer.
- Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. 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. 2011. Backdoors to tractable answer-set programming. In *Proc. IJCAI 2011*, 863–868.
- Fichte, J. K., and Szeider, S. 2013. Backdoors to normality for disjunctive logic programs. In *Proc. AAI 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.
- Gaspers, S., and Szeider, S. 2013. Strong backdoors to bounded treewidth SAT. In *Proc. FOCS 2013*, 489–498. IEEE Computer Society.
- Gaspers, S.; Ordyniak, S.; Ramanujan, M. S.; Saurabh, S.; and Szeider, S. 2013. Backdoors to q-Horn. In *Proc. STACS 2013*, volume 20 of LIPIcs, 67–79. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.
- Gottlob, G., and Szeider, S. 2008. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *Comput. J.* 51(3):303–325.
- Knoblock, C. A. 1994. Automatically generating abstractions for planning. *Artif. Intell.* 68(2):243–302.
- Kronegger, M.; Ordyniak, S.; and Pfandler, A. 2014. Backdoors to planning. In *Proc. AAI 2014*, 2300–2307. AAAI Press.
- Kronegger, M.; Ordyniak, S.; and Pfandler, A. 2015. Supplementary material for the paper “Variable-Deletion Backdoors to Planning”, In *Proc. AAI 2015*. Available at the author’s websites (see e.g., <http://dbai.tuwien.ac.at/staff/kronegger/> and <http://dbai.tuwien.ac.at/staff/pfandler/>).
- 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.
- 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.