# On Weak Stubborn Sets in Classical Planning

**Silvan Sievers**[1] and **Martin Wehrle**
[1]University of Basel, Switzerland
silvan.sievers@unibas.ch

## Abstract

Stubborn sets are a pruning technique for state-space search which is well established in optimal classical planning. In this paper, we show that weak stubborn sets introduced in recent work in planning are actually not weak stubborn sets in Valmari's original sense. Based on this finding, we introduce weak stubborn sets in the original sense for planning by providing a generalized definition analogously to generalized strong stubborn sets in previous work. We discuss the relationship of strong, weak and the previously called weak stubborn sets, thus providing a further step in getting an overall picture of the stubborn set approach in planning.

## 1 Introduction

Planning as search is a leading approach to optimal domain-independent planning. As the induced search spaces generally grow exponentially in the size of the compact planning problem description, additional pruning functions are desired to cut down the search space's size. Pruning functions are supposed to retain at least one (optimal) solution.

Stubborn sets have been introduced as a pruning function for model checking Petri nets [Valmari, 1989], in the form of *strong* and *weak* stubborn sets. In the last decade, stubborn sets have also been investigated for AI planning in various ways [Wehrle and Helmert, 2012; Alkhazraji *et al.*, 2012; Wehrle *et al.*, 2013; Wehrle and Helmert, 2014; Winterer *et al.*, 2017; Al-Khazraji, 2017; Keren *et al.*, 2018; Wilhelm *et al.*, 2018; Schulte, 2018; Gnad *et al.*, 2019; Röger *et al.*, 2020]. As pointed out by Valmari in his original work, weak stubborn sets potentially allow for more pruning than their strong counterpart. However, while the strong version has been considered rather extensively, weak stubborn sets have hardly been analyzed for AI planning so far.

In this work, we revisit weak stubborn sets for planning. We show that recently discussed pruning functions called weak stubborn sets for planning [Winterer *et al.*, 2017; Al-Khazraji, 2017; Wilhelm *et al.*, 2018] are actually not weak stubborn sets in the formal sense as introduced by Valmari [1989]. Motivated by this finding, we introduce weak stubborn sets in Valmari's sense for classical planning. We define them as generalized weak stubborn sets, analogously to generalized strong stubborn sets [Wehrle and Helmert, 2014], and investigate their relationship to existing stubborn set approaches. Altogether, our work provides a further step in getting an overall picture and understanding of the stubborn set approach in AI planning.

## 2 Preliminaries

We consider optimal SAS$^+$ planning, where states of the world are represented based on a finite set of finite-domain variables $\mathcal{V}$. Every variable $v \in \mathcal{V}$ has a finite domain $\mathcal{D}(v)$. For a subset $P \subseteq \mathcal{V}$, a *partial state* $s$ is defined as a function $s : P \to \mathcal{D}(P)$ that maps the variables in $P$ to the domain of their variables, where $\mathcal{D}(P)$ is defined as $\cup_{v \in P} \mathcal{D}(v)$. We say that $s$ is *defined* on variables in $P$. For all remaining variables $\mathcal{V} \setminus P$ that are not contained in $P$, the partial state $s$ is *undefined*. In the special case where $s$ is defined on all variables in $\mathcal{V}$, we call $s$ a *state*. A variable/value assignment $\{v \mapsto p\}$ is called a *fact*. For a partial state $s$ and fact $\{v \mapsto p\}$, we will sometimes use the notation $s \models \{v \mapsto p\}$ to denote that $v$ has value $p$ in $s$. Furthermore, for partial states $s$ and $s'$ and a variable $v$, we say that $s$ and $s'$ *match on* $v$ iff $s(v)$ and $s(v')$ are defined, and $s(v) = s'(v)$. We say that $s$ *satisfies* $s'$ if $s$ and $s'$ match on all variables where $s'$ is defined.

States are changed with *operators* $o = \langle pre(o), \mathit{eff}(o) \rangle$, where $pre(o)$ and $\mathit{eff}(o)$ are partial states. Every operator $o$ has an associate non-negative cost $cost(o) \in \mathbb{R}_0^+$. An operator $o$ is *applicable* in state $s$ iff $s$ satisfies $pre(o)$. In this case, the application of $o$ in $s$ yields the successor state $o(s)$, where $o(s)$ is obtained from $s$ by changing the values of all variables in $s$ such that $o(s)$ satisfies $\mathit{eff}(o)$ and retains the values of the variables in $s$ where $\mathit{eff}(o)$ is undefined. For a state $s$, the set of applicable operators in $s$ is denoted with $app(s)$.

A planning task $\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_\star)$ is defined as a tuple consisting of a finite set of finite-domain variables $\mathcal{V}$, a finite set of operators $\mathcal{O}$, an initial state $s_0$ and a partial goal state $s_\star$. A sequence $\pi$ of operators which is iteratively applicable starting in a state $s$ and which leads to a state $s'$ such that $s'$ satisfies the partial goal state is called a *plan* for $s$. A plan for $\Pi$ is a plan for $s_0$ of $\Pi$. States for which a plan exists are called *solvable*, other states are called *unsolvable*. The *cost* of a plan is the summed costs of each operator of the sequence. A plan $\pi$ for $s$ is called *optimal* if its cost is minimal among all plans for $s$. A plan $\pi$ for $s$ is called *strongly optimal* if $\pi$ is optimal for $s$ and contains a minimal number of zero-cost

operators among all optimal plans for $s$. Our objective is to find an optimal plan for $\Pi$.

## 2.1 Safe Pruning with Stubborn Sets

We consider planning as search using *safe pruning functions* that restrict the successors of a state $s$ to subsets of $app(s)$ as introduced by Wehrle and Helmert [2014]. A pruning function is called *safe* if for every state $s$, the costs of an optimal plan for $s$ in the pruned state space and in the original state space are the same.

As a basis for the paper, we provide the notions of generalized strong stubborn sets. Stubborn sets crucially rely on the notion of *interference*. In previous work, interference has been defined both in a state-based fashion [e.g., Wehrle and Helmert, 2014], as well as in a syntax-based fashion as an approximation thereof [e.g., Alkhazraji *et al.*, 2012]. As the differentiation will be crucial for the analyses in the following, we will introduce both the state-based notions (now) and syntactic-based notions (later in the paper), using the naming convention to leave out the name "state-based" for brevity. We start by giving the definition of disabling operators and interference according to Wehrle and Helmert [2014].

**Definition 1** (disabling, conflicting)**.** *Let $o_1$ and $o_2$ be operators of a planning task $\Pi$, let $s$ be a state of $\Pi$, and let $o_1$ and $o_2$ be both applicable in $s$. Then we say that*

- $o_1$ disables $o_2$ in $s$ *if* $o_2 \notin app(o_1(s))$.
- $o_1$ disables $o_2$ on fact $\{v \mapsto p\}$ in $s$ *if* $pre(o_2) \models \{v \mapsto p\}$ *and* $o_1(s) \not\models \{v \mapsto p\}$.
- $o_1$ and $o_2$ conflict in $s$ *if* $s_{12} = o_2(o_1(s))$ *and* $s_{21} = o_1(o_2(s))$ *are both defined and differ (i.e.,* $s_{12} \neq s_{21}$*).*

Two operators *interfere* in a state if they cannot be executed in both possible orderings, leading to the same state. We distinguish interference and weak interference.

**Definition 2** (interference)**.** *Let $o_1$, $o_2$ be operators of a planning task $\Pi$, and let $s$ be a state of $\Pi$. We say that* $o_1$ weakly interferes *with $o_2$ in $s$ if they are both applicable in $s$, and*

- *$o_1$ disables $o_2$ in $s$, or*
- *$o_1$ and $o_2$ conflict in $s$.*

*Furthermore, we say that* $o_1$ interferes *with $o_2$ in $s$ if they are both applicable in $s$, and*

- *$o_1$ weakly interferes with $o_2$ in $s$, or*
- *$o_2$ disables $o_1$ in $s$.*

The differentiation of weak interference and interference will be important for the differentiation of strong and weak stubborn sets. We observe that the notion of interference is symmetric (conflict is a symmetric notion and two operators interfere if one disables the other, no matter the direction), but that this is not the case for weak interference (i.e., $o_1$ can weakly interfere with $o_2$ in a state $s$, while $o_2$ does not necessarily weakly interfere with $o_1$ in $s$). Also note that weak interference implies interference.

Furthermore, for the definition of generalized strong stubborn sets, we need the notion of necessary enabling sets.

**Definition 3** (necessary enabling set, Wehrle and Helmert 2014)**.** *Let $\Pi$ be a planning task, let $o$ be an operator of $\Pi$, and let Seq be a set of operator sequences applicable in the initial state of $\Pi$. A* necessary enabling set *for $o$ and Seq is a set $N$ of operators such that every operator sequence in Seq which includes $o$ as one of its operators also includes an operator $o' \in N$ before the first occurrence of $o$.*

We now provide the conditions needed for generalized strong stubborn sets, introduced by Wehrle and Helmert [2014]. For a planning task $\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_\star)$, a solvable state $s$ in $\Pi$, and an associated *envelope* $E \subseteq \mathcal{O}$, we define the task $\Pi_s^E = (\mathcal{V}, E, s, s_\star)$.[1] For such a task $\Pi_s^E$, we define $Opt$ as the set of strongly optimal plans for $\Pi_s^E$, and $\mathcal{S}_{Opt}$ as the set of states that are visited by at least one strongly optimal plan in $Opt$. Let $T \subseteq \mathcal{O}$ be a set of operators. We define the following conditions for $E$ and $T$:

C1: $E$ includes all operators from at least one strongly optimal plan for $s$ (in task $\Pi$).

C2: $T$ contains at least one operator from at least one strongly optimal plan for $\Pi_s^E$.

C3: For every $o \in T$ not applicable in $s$, $T$ contains a necessary enabling set for $o$ and $Opt$.

C4: For every $o \in T$ applicable in $s$, $T$ contains all operators $o' \in E$ such that $o'$ interferes with $o$ in any state $s' \in \mathcal{S}_{Opt}$.

We can now define generalized strong stubborn sets.

**Definition 4** (generalized strong stubborn set, Wehrle and Helmert 2014)**.** *Let $\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_\star)$ be a planning task, $E$ an associated envelope, and $s$ a state of $\Pi$. A* generalized strong stubborn set (GSSS) in $s$ *is a set of operators $T \subseteq \mathcal{O}$ with the following properties. If $s$ is an unsolvable state or a goal state, every set $T \subseteq \mathcal{O}$ is a GSSS in $s$. If $s$ is a solvable non-goal state, then $T$ is a GSSS in $s$ if conditions C1, C2, C3 and C4 are true for $E$ and $T$.*

Applying only the operators from a GSSS in a state $s$ guarantees that for at least one strongly optimal plan $\pi$ in $s$, there is a permutation $\pi'$ of $\pi$ such that $\pi'$ is not pruned in $s$ and $\pi'$ is a plan in $s$. Hence, GSSS yield safe pruning functions.

As computing the interference relationship effectively relies on computing the set of states in $\mathcal{S}_{Opt}$, which is computationally as hard as solving the planning task itself, this computation has been approximated in practice. A simple approximation is checking the interference based on the pure operator syntax (we will come back to this in the next section). A more sophisticated approximation for both $\mathcal{S}_{Opt}$ and the operator interference relation is to use *mutex*-based interference. Two facts $f$ and $f'$ are called mutex if no reachable state satisfies both $f$ and $f'$ [Bonet and Geffner, 2001]. While the general computation of mutexes is as hard as solving the original planning task, there exist sound but incomplete polynomial-time algorithms to compute mutexes up to a fixed size [e.g., Rintanen, 2008; Helmert, 2009]. Two operators $o$ and $o'$ are called mutex if there exists a pair of facts

[1] Wehrle and Helmert [2014] characterize an envelope as "an operator set that is known to be sufficient in the sense that we can safely treat all operators outside the envelope as if they did not exist".

$(f, f')$ such that $f$ and $f'$ are mutex, $f$ is contained in the precondition of $o$, and $f'$ is contained in the precondition of $o'$. For mutex operators, we know that there is no reachable state where both of them are applicable, hence in particular they do not interfere in any state in $\mathcal{S}_{Opt}$.

## 3 The Operator Shifting Property

In his original work, Valmari [1989] stated that stubborn sets $T$ in a state $s$ remain stubborn sets in the successor state $o(s)$ if $o$ is not contained in $T$ (see Theorem 1.19 in his paper). While this stubborn property holds for strong and weak stubborn sets, it does no longer hold for GSSS, as GSSS restrict the consideration to preserve only *at least one* strongly optimal plan (instead of preserving a permutation of *every* plan) for $s$: If there are strongly optimal plans $\pi_1$ and $\pi_2$ for $s$, and a GSSS $T$ only contains operators in $\pi_1$, then $T$ does not need to be a GSSS in $o(s)$ if $o$ is the first operator of $\pi_2$.[2]

In the following, we focus on a slightly more general property to characterize stubborn sets, which reflects the original key idea of stubborn sets on the one hand, but abstracts away from the set of plans that are preserved. We call this property "operator shifting property".

**Definition 5** (operator shifting property)**.** *Let $\Pi$ be a planning task, $s$ be a state of $\Pi$ and $T$ be a subset of the operators of $\Pi$. We say that $T$ has the* operator shifting property *in $s$ if for all plans $\pi = o_1 \dots o_n$ for $s$ with $\{o_1, \dots, o_n\} \cap T \neq \emptyset$, the following holds, assuming $o_i$ to be the operator with minimal index that is contained in $T$ (i.e., $o_1, \dots o_{i-1} \notin T$):*

1. *$o_i$ can be shifted to the front of $\pi$, i.e., $\pi' = o_i o_1 \dots o_{i-1} o_{i+1} \dots o_n$ is a plan, and*
2. *$o_i$ is also applicable in all intermediate states $o_j(\dots(o_1(s))\dots)$ for $j = 1, \dots, i-1$.*

Valmari [1989] already showed in the derivation of stubborn sets that this property holds for both strong and weak stubborn sets. Later on, Wehrle and Helmert [2014] showed in the safety proof of GSSS that the operator shifting property also holds for GSSS. However, as we will see in the following, the recently considered pruning techniques in planning which have been called weak stubborn sets, including the approaches by Winterer *et al.* [2017], Al-Khazraji [2017], and Wilhelm *et al.* [2018], do *not* satisfy (the second part of) the operator shifting property. All of these recent techniques use notions of interference solely based on syntactically checking preconditions and effects of operators. We restate their definitions in the following.

**Definition 6** (syntactic weak interference)**.** *Let $o_1$, $o_2$ be operators of a planning task $\Pi$. We say that $o_1$* syntactically weakly interferes *with $o_2$ if there are facts $f_1 = \{v \mapsto p\}$ and $f_2 = \{v \mapsto p'\}$ with $p \neq p'$ such that $\mathit{eff}(o_1) \models f_1$ and*

- *$\mathit{pre}(o_2) \models f_2$, or*
- *$\mathit{eff}(o_2) \models f_2$.*

*Furthermore, $o_1$* syntactically interferes *with $o_2$ if it syntactically weakly interferes with $o_2$ or $\mathit{eff}(o_2) \models f_2$ and $\mathit{pre}(o_1) \models f_1$.*

Using interference approximations like Def. 6 is sound because every superset of interference relations still yields a safe pruning function, to the expense of less pruning power. For our later analysis, we restate the definition of weak stubborn sets along the lines of the definition used by Winterer *et al.* [2017], Al-Khazraji [2017], Wilhelm *et al.* [2018]. As we will see that these sets are not weak stubborn sets in the sense of Valmari, we redefine their name and call them "compliant stubborn sets" in the following. The definition needs the further notion of *disjunctive action landmarks* for a state $s$ [Helmert and Domshlak, 2009], i.e., sets of operators $L$ such that every plan for $s$ contains at least one operator in $L$.

**Definition 7** (compliant stubborn sets)**.** *Let $\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_\star)$ be a planning task, and let $s$ be a state of $\Pi$. A* compliant stubborn set (CSS) in $s$ *is a set of operators $T \subseteq \mathcal{O}$ with the following properties. If $s$ is an unsolvable state or a goal state, every set $T \subseteq \mathcal{O}$ is a CSS in $s$. If $s$ is a solvable non-goal state, then the following conditions must be true for $T$ to be a CSS in $s$:*

1. *$T$ contains a disjunctive action landmark for $s$.*
2. *For every $o \in T$ that is not applicable in $s$, $T$ contains a necessary enabling set for $o$ and the set of all applicable operator sequences in $s$.*
3. *For every $o \in T$ that is applicable in $s$, $T$ contains all operators $o'$ such that $o$ syntactically weakly interferes with $o'$.*

In contrast to GSSS, the condition for applicable operators is simplified to syntactic weak interference. As pointed out in the literature, CSS yield safe pruning functions. However, they do not satisfy the operator shifting property.

**Theorem 1.** *Compliant stubborn sets do not satisfy the operator shifting property.*

*Proof.* We provide an example to show that Def. 5 is not satisfied. Consider the planning task with initial state $s_0 = \{v \mapsto 0, X \mapsto 0, Y \mapsto 0, Z \mapsto 0\}$, goal $s_\star = \{X \mapsto 1, Y \mapsto 1, Z \mapsto 1\}$, and operators $o_1, o_2, o_3$ ($\mathit{cost} = 1$) with

- $\mathit{pre}(o_1) = \{v \mapsto 0\}$, $\mathit{eff}(o_1) = \{v \mapsto 1, X \mapsto 1\}$
- $\mathit{pre}(o_2) = \{v \mapsto 1\}$, $\mathit{eff}(o_2) = \{v \mapsto 0, Y \mapsto 1\}$
- $\mathit{pre}(o_3) = \{v \mapsto 0\}$, $\mathit{eff}(o_3) = \{Z \mapsto 1\}$

[2] To see this, consider the planning task with initial state $s_0 = \{v_i \mapsto 0\} \cup \{G \mapsto 0\}$ for $1 \leq i \leq 6$, goal $s_\star = \{G \mapsto 1\}$, and operators $o_i$ with $\mathit{pre}(o_i) = \{v_i \mapsto 0\}$ and $\mathit{eff}(o_i) = \{v_i \mapsto 1\}$ for $i = 1, \dots, 4$, and operators $o_5$ and $o_6$ with $\mathit{pre}(o_5) = \{v_1 \mapsto 1, v_2 \mapsto 1\}$, $\mathit{pre}(o_6) = \{v_3 \mapsto 1, v_4 \mapsto 1\}$ and $\mathit{eff}(o_j) = \{G \mapsto 1\}$ for $j = 5, 6$, all with costs of 1. Note that we generally use upper case letters to denote variables mentioned in the goal. In $s_0$, one can choose to use $o_1, o_2$ and $o_5$ or alternatively $o_3, o_4$ and $o_6$ to optimally achieve the goal. For example, $T = \{o_1\}$ is a GSSS in $s_0$ because it contains an operator from a strongly optimal plan for $s_0$, and all interfering operators are contained as well (as there are none of those). However, as $o_3$ starts a strongly optimal plan in $s_0$ as well, $T$ is no longer a GSSS in $s' := o_3(s_0)$ because $o_1, o_2, o_5$ yield a plan of higher cost for $s'$ than $o_4, o_6$ (and hence, $T$ does not contain an operator from a strongly optimal for $s'$).

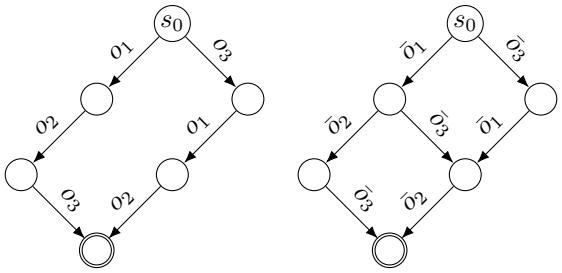


Figure 1: Compliant vs. stubborn sets

The set $T = \{o_3\}$ is a CSS in $s_0$: $T$ is a disjunctive action landmark in $s_0$ and the only applicable operator $o_3$ does not syntactically weakly interfere with $o_1$ or $o_2$. Hence neither $o_1$ nor $o_2$ needs to be included in $T$.

We observe that there are two plans for $s_0$, namely $\pi_1 = o_1 o_2 o_3$ and $\pi_2 = o_3 o_1 o_2$, as illustrated in the left part of Fig. 1. $T$ does not satisfy the operator shifting property in $s_0$ because condition 2. of Def. 5 is not satisfied for $\pi_1$: Although $o_3$ can be shifted to the front, yielding the plan $\pi_2$, $o_3$ is *not* applicable in the intermediate state $o_1(s_0)$. □

The example in the proof of Theorem 1 illustrates a key difference of stubborn sets and compliant stubborn sets: Stubborn sets satisfy the operator shifting property, yielding a state space as shown in the right part of Fig. 1. In contrast, compliant stubborn sets only allow applying operators from $T$ in certain (but generally not all) intermediate states, as there can be "blocks" of operators that (temporarily) disallow applying operators from $T$ within such a block, yielding a state space as shown in the left part of Fig. 1.

From a more technical point of view, the operator shifting property does not hold for CSS because for applicable operators in a CSS, the only requirement to include additional operators is syntactic weak interference (see bullet point 3 in Def. 7). To reflect the original idea of weak stubborn sets in Valmari's sense, additional operators need to be included in this case [1989]. The definition of (generalized) weak stubborn sets addresses this point.

## 4 Generalized Weak Stubborn Sets

Motivated by the results of the previous section, we formally adapt Valmari's weak stubborn sets for classical planning in the following. We provide a generalized version thereof, analogously to the generalization of strong stubborn sets [Wehrle and Helmert, 2014]. For the definition of generalized weak stubborn sets, we additionally need the notion of *enabling*.

**Definition 8** (enabling)**.** *Let $o_1$ and $o_2$ be operators of a planning task $\Pi$, let $s$ be a state of $\Pi$. Let $o_1$ be applicable in $s$, and $o_2$ be* not *applicable in $s$. Then we say that*

- *$o_1$* enables *$o_2$ in $s$ if $o_2 \in app(o_1(s))$.*
- *$o_1$* enables *$o_2$* on fact *$\{v \mapsto p\}$ in $s$ if $s \not\models \{v \mapsto p\}$, $o_1(s) \models \{v \mapsto p\}$, and $pre(o_2) \models \{v \mapsto p\}$.*

Note that enabling $o$ on a precondition fact $p$ in $s$ is a "local" property in the sense that there is no conclusion on the applicability of $o$ in $s$ (in contrast to disabling $o$ on $p$ in $s$).

In the following definition of generalized weak stubborn sets, we refer to conditions C1–C3 stated in Section 2.

**Definition 9** (generalized weak stubborn set)**.** *Let $\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_\star)$ be a planning task, $E$ an associated envelope, and $s$ a state of $\Pi$. A* generalized weak stubborn set *(GWSS) in $s$ is a set of operators $T \subseteq \mathcal{O}$ with the following properties. If $s$ is an unsolvable state or a goal state, every set $T \subseteq \mathcal{O}$ is a GWSS in $s$. If $s$ is a solvable non-goal state, then $T$ is a GWSS in $s$ if conditions C1, C2, C3 and the following C4' are true for $E$ and $T$.*

*C4': For all $o \in T$ applicable in $s$, $T$ contains all $o' \in E$ such that $o$ weakly interferes with $o'$ in any state $s' \in \mathcal{S}_{Opt}$. Additionally, for all $\{v \mapsto p\}$ with $pre(o) \models \{v \mapsto p\}$:*

- *$T$ contains all operators $o'$ such that $o'$ disables $o$ on fact $\{v \mapsto p\}$ in any state $s' \in \mathcal{S}_{Opt}$, or*
- *$T$ contains all operators $o'$ such that $o'$ enables $o$ on fact $\{v \mapsto p\}$ in any state $s' \in \mathcal{S}_{Opt}$.*

The central difference to GSSS is that GWSS $T$ allow for a relaxed constraint on the applicable operators in $T$, i.e., they generalize condition C4 to C4': C4' considers weak interference instead of interference and additionally requires, for every precondition fact $p$ of an applicable operator in $T$, that $T$ contains either all "disablers" of $p$ or all "enablers" of $p$. Always including all disablers yields GSSS (C4' = C4 in this case) because including disabling operators means including interfering operators. Alternatively, GWSS also allow including all enablers of $p$ instead.

We remark that in contrast to the definition of GSSS, which is purely based on the semantics of the planning task, the definition of GWSS additionally relies on a fact-based, and thus syntax-based, notion of enabling and disabling (cf. Def. 8). While defining GWSS purely based on the semantics would be possible by using only the state-based notion of enabling and disabling, such a definition would no longer allow for including both enablers and disablers for different precondition facts of the same operator, thus resulting in a less general definition. As the original definition of weak stubborn sets by Valmari [1989] already allowed choosing enablers or disablers on a fact basis, we define GWSS in the same style.

The correctness proof for GWSS works analogously to the corresponding proof for GSSS [Wehrle and Helmert, 2014].

**Theorem 2.** *Let $T$ be a GWSS in $s$. Then the successor pruning function defined as $succ(s) := T \cap app(s)$ is safe.*

*Proof.* Let $s$ be a solvable non-goal state and $T$ be a GWSS in $s$. With the same proof arguments of Wehrle and Helmert, due to C1, strongly optimal plans for $\Pi_s^E$ are also strongly optimal for state $s$ in $\Pi$. Hence it suffices to show that $T$ contains the first operator of a strongly optimal plan for $\Pi_s^E$.

Consider a strongly optimal plan $\pi = o_1, \ldots, o_n$ for $\Pi_s^E$ where at least one of $\pi$'s operators is also contained in $T$. Such a plan must exist because of C2. Let $k \in \{1, \ldots, n\}$ be the minimal index for which $o_k \in T$, and let $s^0, \ldots, s^n$ be the sequence of states that are visited by $\pi$, such that $s^0 = s$, and $s^i = o_i(s^{i-1})$ for all $1 \leq i \leq n$. Using the same proof arguments as Wehrle and Helmert, we observe the following:

- The states $s^0, \ldots, s^n$ are contained in $\mathcal{S}_{Opt}$ because $\pi$ is strongly optimal.

- The operator $o_k$ is applicable in $s$: Otherwise, because of C3, $T$ would contain a necessary enabling set $N$ for $o_k$ and $Opt$. By the definition of necessary enabling sets and because $\pi \in Opt$, $\pi$ would include an operator from $N$ before $o_k$, thus contradicting the minimality of $o_k$.
- The operator $o_k$ does not weakly interfere with any of the operators $o_1, \dots, o_{k-1}$ in any of the states $s^j$ for $0 \leq j \leq k-1$: Otherwise, the weakly interfering operators would be contained in $T$ because of C4' with $o = o_k$, which would contradict the minimality of $k$.

We show that if $o_k$ is not already the first operator in $\pi$ (i.e., for $k > 1$), it can be shifted to the front of $\pi$. We already know that $o_k$ is applicable in $s^0$, and that $o_1$ is applicable in $s^0$ (otherwise $\pi$ would not be applicable in $s$).

We observe that $o_k$ must also be applicable in $s^1$: Assume the opposite, i.e., assume $o_1$ disables $o_k$ in $s^0$. According to C4', as $o_k$ is applicable in $s = s^0$, for all precondition facts $p$ of $o_k$, either all disablers or all enablers for $p$ in some state in $\mathcal{S}_{Opt}$ are contained in $T$, hence in particular all disablers or enablers for $p$ in states $s^0, \dots, s^{k-1}$.

- Case "all disablers in $T$": If all disablers of $p$ (in particular in $s^0$) are contained in $T$, then $T$ must contain $o_1$, contradicting the minimality of $k$. Hence in this case, $o_1$ cannot disable $o_k$.
- Case "all enablers in $T$": If $o_1$ disables $o_k$ in $s^0$, then $o_1$ disables $o_k$ on some precondition fact $p$ in $s^0$. As $o_k$ is applicable in $s^{k-1}$ (because $o_k$ is part of $\pi$), it follows that there must be an operator among $o_2, \dots, o_{k-1}$ that enables $o_k$ on $p$ again in some state $s^2, \dots, s^{k-2}$ (otherwise $o_k$ would not be applicable in $s^{k-1}$ in $\pi$). However, as all enablers of $p$ in states $s^1, \dots, s^{k-2}$ are already contained in $T$, this contradicts the minimality of $k$.

Overall, we observe that $o_k$ is applicable in $s^1$. With the same proof arguments of Wehrle and Helmert, this argument can be repeated to show that $o_{k-1}$ and $o_k$ can be swapped in $\pi$, still yielding a valid plan because $o_k$ does not weakly interfere with $o_{k-1}$. Finally, this argument in turn can be repeated to swap $o_k$ to position $k-2$, $k-3$, ..., until we get the plan $\pi' = o_k, o_1, \dots, o_{k-1}, o_{k+1}, \dots, o_n$. We observe that $\pi'$ is a permutation of $\pi$ for which the first operator is contained in $T$. This concludes the proof. □

The main difference to the proof of Wehrle and Helmert is the argument why $o_k$ can be shifted to the front of $\pi$, still being applicable in all intermediate states $s^1, \dots, s^{k-1}$. Informally, if applicable in $s^0$, then $o_k$ must stay enabled because it cannot be disabled, either because all disablers are already contained in $T$, or if $o_k$ were disabled, then it could not be enabled once again because all enablers are contained in $T$. Hence it is not possible that $o_k$ gets disabled and then enabled again later in $s^1, \dots, s^{k-1}$. In particular, this means that $T$ satisfies the operator shifting property in $s$ (Def. 5).

We conclude the section with the following result.

**Theorem 3.** *The pruning power of GWSS is exponentially higher than the pruning power of GSSS.*

*Proof.* We extend Example 2 by Al-Khazraji [2017, Section 4.1.2] which applies for CSS but not directly for GWSS. For $n \geq 1$, consider the planning task with variables $\mathcal{V} = \{x, y, G_1, G_2\} \cup \{v_i \mid 1 \leq i \leq n\}$, initial state $s_0 = \{V \mapsto 0 \mid V \in \mathcal{V}\}$, goal $s_\star = \{G_1 \mapsto 1, G_2 \mapsto 1\}$, and operators $o_1, o_2, o_3, \hat{o}_i, \hat{o}'_i$ for $1 \leq i \leq n$ ($cost = 1$) with

- $pre(o_1) = \{x \mapsto 0\}$, $\mathit{eff}(o_1) = \{G_1 \mapsto 1\}$
- $pre(o_2) = \{x \mapsto 0\}$, $\mathit{eff}(o_2) = \{G_2 \mapsto 1\}$
- $pre(o_3) = \{x \mapsto 0\}$, $\mathit{eff}(o_3) = \{x \mapsto 1, y \mapsto 1\}$
- $pre(\hat{o}_i) = \{v_i \mapsto 0\}$, $\mathit{eff}(\hat{o}_i) = \{y \mapsto 1, v_i \mapsto 1\}$
- $pre(\hat{o}'_i) = \{v_i \mapsto 1\}$, $\mathit{eff}(\hat{o}'_i) = \{y \mapsto 0, v_i \mapsto 0\}$.

For both GWSS and GSSS, we use envelopes containing all operators to satisfy C1 and we use disjunctive action landmarks to satisfy C2.

In $s_0$, both $T_1 = \{o_1\}$ and $T_2 = \{o_2\}$ are disjunctive action landmarks. We argue that both $T_1$ and $T_2$ already satisfy the conditions of being GWSS in $s_0$: $o_1$ and $o_2$ are applicable in $s_0$ and thus, to satisfy C4', we need to include all operators $o$ such that $o_1$ or $o_2$ weakly interfere with $o$. However, neither $o_1$ nor $o_2$ conflict with or disable any other operator in the task. Furthermore, we can choose to include all enablers (rather than all disables) of the precondition $\{x \mapsto 0\}$ of $o_1$ and $o_2$. As there are no such enablers, we do not need to include any further operators and we conclude that both $T_1$ and $T_2$ are GWSS in $s_0$. The pruned state space with either of these GWSS contains 3 states (assuming no successors of goal states are considered): $s_0$, one intermediate state where either $G_1$ or $G_2$ is set to 1 depending on using $T_1$ or $T_2$, and a goal state where both $G_1$ and $G_2$ are set to 1.

In contrast, GSSS do not allow the option to include the enablers, but require including all disablers of $\{x \mapsto 0\}$ instead: Recall that C4, in contrast to C4', requires including all operators $o$ which interfere with $o_1$ or $o_2$, which translates to including all operators $o$ which disable $o_1$ and $o_2$ in any state, or, in other words, for any precondition of $o_1$ and $o_2$. (There are no operators conflicting with $o_1$ or $o_2$, and if there were, they would need to be included in the case of GWSS, too.) Therefore, any GSSS $T$ in $s_0$ must not only include $o_1$ or $o_2$, but also include $o_3$ because $o_3$ disables $o_1$ and $o_2$ in $s_0$ on $\{x \mapsto 0\}$. This directly leads to the inclusion of $o_1$ or $o_2$, depending on which of both we start with. Furthermore, since $o_3$ is applicable in $s_0$, $T$ must also include $\hat{o}'_i$ for $1 \leq i \leq n$ because they conflict with $o_3$. As the operators $\hat{o}'_i$ are not applicable in $s_0$, $T$ must also include necessary enabling sets for them: since every operator $\hat{o}_i$ is a necessary enabling set for $\hat{o}'_i$ for $1 \leq i \leq n$ due to its effect $v_i \mapsto 1$, $T$ also includes all $\hat{o}_i$. As a consequence, $T$ contains all operators and there is no pruning obtained. Since $\hat{o}_i$ and $\hat{o}'_i$ can be applied in any permutation of $\langle 1, \dots, n \rangle$, leading to different intermediate states, the state space of the task grows exponentially in $n$. Thus GSSS generate a state space exponentially larger than GWSS, concluding the proof. □

## 5 Relationship To Compliant Stubborn Sets

We analyze the relationship of GWSS and CSS. Although having similar flavors, GWSS and CSS have orthogonal properties (called R1 and R2 in the following), which influence their *pruning power*, i.e., how much they can prune the set of applicable operators in a state.

- R1: CSS are *stricter* than GWSS: CSS can only be applied with syntactic, not with state-based interference. This has already been noted by Winterer *et al.* [2017].
- R2: CSS are *less restrictive* than GWSS: CSS do not satisfy the operator shifting property, as operators in a CSS need not be applicable in all intermediate states of operator sequences (recall the left part of Fig. 1).

In the following, we consider the pruning power of GWSS and CSS more closely. For the special case when GWSS are applied using the *syntactic interference relation* for C4' like CSS, the same disjunctive action landmarks as CSS, and full envelopes, CSS yield subsets compared to GWSS because GWSS include the same operators as CSS but additionally enablers or disablers of applicable operators to satisfy C4'. Hence, CSS have a higher pruning power in this case.

In contrast, in the general case when GWSS are applied with state-dependent interference relations (which cannot be done for CSS), there is no dominance relation.

**Theorem 4.** *GWSS and CSS are incomparable in terms of pruning power.*

*Proof.* To see that CSS can yield more pruning than GWSS, consider again the example in the proof of Theorem 1. As we have already seen there, $T = \{o_3\}$ is a CSS in $s_0$ because $T$ is a disjunctive action landmark in $s_0$ and $o_3$ does not syntactically weakly interfere with $o_1$ or $o_2$. Therefore, the state space when using $T$ consists of the linear chain of the following 4 states, which we denote in the format $\langle v, X, Y, Z\rangle$: $\langle 0,0,0,0\rangle$, $\langle 0,0,0,1\rangle$, $\langle 1,1,0,1\rangle$, $\langle 0,1,1,1\rangle$.

In contrast, we show that every GWSS $T$ in $s_0$, using the full envelope $E = \{o_1, o_2, o_3\}$, contains at least both applicable operators $o_1$ and $o_3$. We observe that all three operators are necessary in any strongly optimal plan. If we choose $o_1$ to satisfy C2, we must include $o_3$ because $o_1$ disables $o_3$ in $s_0$ and therefore weakly interferes with it. If, however, we choose $o_3$ to satisfy C2, there are no operators with which $o_3$ weakly interferes, but we must either include all enablers or disablers of its precondition $\{v \mapsto 0\}$. If we choose all disablers, we have to include $o_1$ and we are done. Otherwise, if we choose all enablers, we have to include $o_2$. Since $o_2$ is not applicable in $s_0$, we have to include $o_1$ as a necessary enabling set for $o_2$ and the set of all operators. We conclude by observing that using the minimal GWSS $T = \{o_1, o_3\}$ in $s_0$ yields a state space that additionally contains the states $\langle 1,1,0,0\rangle$, $\langle 0,1,1,0\rangle$ and $\langle 1,1,1,0\rangle$ compared to CSS.

To see that GWSS can yield more pruning than CSS, consider the planning task with initial state $s_0 = \{v \mapsto 0, w \mapsto 0, G_1 \mapsto 0, G_2 \mapsto 0\}$, goal $s_\star = \{G_1 \mapsto 1, G_2 \mapsto 1\}$, and operators $\mathcal{O} = \{o_1, o_2, o_3\}$ ($cost = 1$) with

- $pre(o_1) = \{v \mapsto 0\}$, $\mathit{eff}(o_1) = \{w \mapsto 1, G_1 \mapsto 1\}$
- $pre(o_2) = \{v \mapsto 1\}$, $\mathit{eff}(o_2) = \{w \mapsto 2, G_2 \mapsto 1\}$
- $pre(o_3) = \top$, $\mathit{eff}(o_3) = \{v \mapsto 1\}$.

We denote states in the format $\langle v, w, G_1, G_2\rangle$. Consider GWSS with envelope $E = \mathcal{O}$ and the policy to include all enablers for every precondition fact of an applicable operator. We show that $T = \{o_1\}$ is a GWSS for $s_0$. Apparently, choosing $o_1$ satisfies C2. As $o_1$ is applicable in $s_0$, $T$ has to include all operators $o$ such that $o_1$ weakly interferes with $o$ in any state. However, $o_1$ does not weakly interfere with $o_2$ in any state because they cannot be both applicable in any state (they have mutex preconditions), and $o_1$ does not weakly interfere with $o_3$ because $o_3$ has no precondition and they do not conflict. To satisfy C4', $T$ must additionally include all enablers of $o_1$; however, there are none. Hence, $T$ is a GWSS for $s_0$, yielding the state space which consists of the linear chain $\langle 0,0,0,0\rangle$, $\langle 0,1,1,0\rangle$, $\langle 1,1,1,0\rangle$, $\langle 1,2,1,1\rangle$.

In contrast, every CSS $T$ in $s_0$ must contain all applicable operators: When starting with $T = \{o_1\}$, this is the case because $o_1$ and $o_2$ have conflicting effects and are hence both included, so $T$ must also include $o_3$ as a necessary enabling set for $o_2$. Analogous arguments hold when starting with $T = \{o_2\}$, and when starting with $T = \{o_3\}$, $T$ needs to include $o_1$ due to syntactic weak interference. We conclude by observing that using CSS yields the additional states $\langle 1,0,0,0\rangle$ and $\langle 1,2,0,1\rangle$ compared to GWSS. □

Overall, we observe that properties R1 and R2 orthogonally determine the pruning power of GWSS and CSS. We think that they deserve further investigation in the future, allowing potentially for more powerful pruning techniques. We will come back to this point in the conclusions.

## 6 Evaluation

To complete the picture, we investigate the pruning power of GWSS, CSS, and GSSS within an A$^*$ search also in practice. We use the state-of-the-art saturated cost partitioning (SCP) heuristic [Seipp *et al.*, 2020] computed over pattern databases [Edelkamp, 2001], generated systematically up to pattern size 2 and via hill climbing [Haslum *et al.*, 2007], and over Cartesian abstractions [Seipp and Helmert, 2018].[3]

We added implementations of CSS and GWSS to the existing implementation of GSSS in the planner Fast Downward 20.06 [Helmert, 2006]. As we have seen, GWSS and CSS are incomparable in their pruning power, because CSS cannot be applied with state-based interference, but GWSS need to include additional operators compared to CSS. A particular objective of the experiments is to investigate the pruning power with respect to this property. Hence we use the mutex-based interference for GWSS and GSSS as a state-based approximation. To maximally distinguish GWSS from GSSS, we always choose to include all enablers rather than all disablers to satisfy condition C4' of Def. 9. As the comparison needs a common implementation basis, we use the operator-centric algorithm [Alkhazraji *et al.*, 2012; Wehrle and Helmert, 2014] rather than the more efficient, state-of-the-art atom-centric algorithm as the latter is not compatible with mutex-based interference [Röger *et al.*, 2020]. [4]

Our evaluation uses the STRIPS planning benchmarks of all optimal tracks of all International Planning Competitions

[3] The comparison of pruning power with the blind and LM-cut [Helmert and Domshlak, 2009] heuristics yield qualitatively similar results; see the technical report [Sievers and Wehrle, 2021b].

[4] We also evaluated GSSS and GWSS without mutex-based interference in the atom-centric algorithm and found qualitatively similar results; see the technical report [Sievers and Wehrle, 2021b].

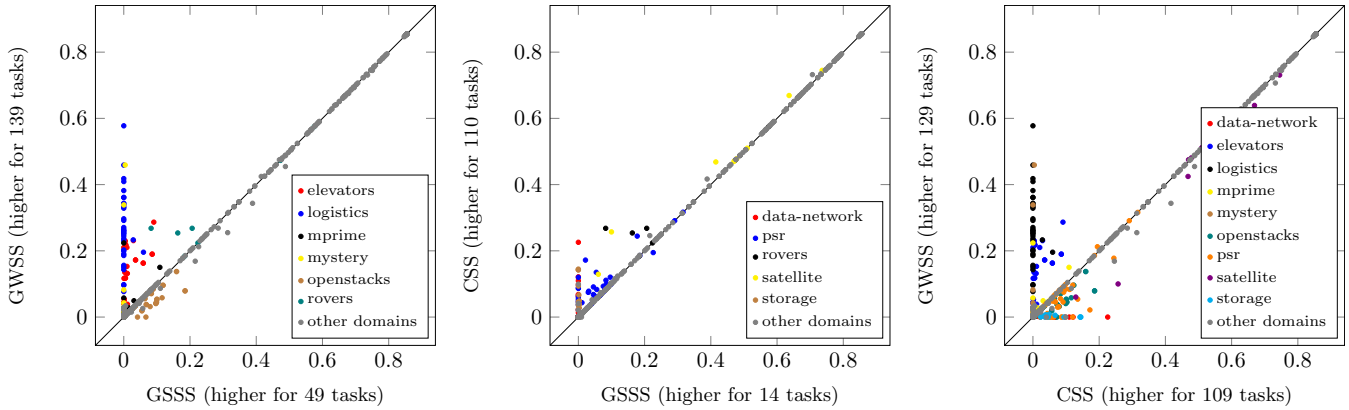


Figure 2: Pruning ratio of GSSS vs. GWSS (left), GSSS vs. CSS (middle), CSS vs. GWSS (right), all with the operator-centric algorithm and mutex-based interference except with CSS. Highlighted domains: difference of pruning ratio above 0.1 for at least one task.

from 1998 to 2018 which yields a set consisting of 1827 tasks from 65 domains. We conduct experiments on Intel Xeon Silver 4114 CPUs using Downward Lab [Seipp *et al.*, 2017] and impose a memory limit of 3.5 GiB and 30 minutes on every planner run. The code, benchmarks and experimental data are published online [Sievers and Wehrle, 2021a].

To assess the pruning power, we define the *pruning ratio* of a (completed) search using a pruning function as follows. We sum up, over all expanded states $s$, the number of *all* successors of $s$ as $n_{all}$ and the number of *generated* successors of $s$ (i.e., successors not pruned by the pruning function) as $n_{gen}$. We define the pruning ratio as $1 - \frac{n_{gen}}{n_{all}}$, yielding values from 0 to 1, where 0 means that no states are pruned and 1 means that all states are pruned. Figure 2 shows scatter plots comparing the pruning ratio of all three methods, highlighting domains where the difference of the pruning ratio is above 0.1.

The results confirm the orthogonal pruning power of the methods. At the left, we compare GSSS, the previous standard in classical planning, against GWSS, the generalization introduced in this work. We observe that GWSS leads to more pruning than GSSS in 139 tasks while the converse is only true in 49 tasks. As GSSS and GWSS use the same mutex-based approximation of state-based interference and only differ in including enablers (GWSS) compared to disablers (GSSS) of applicable operators, we conclude that the former is overall more favorable in terms of pruning power. Using more sophisticated strategies for deciding when to include all disablers or all enablers could possibly increase the pruning power further. The middle plot compares GSSS against CSS. We observe that CSS, like GWSS, lead to more pruning compared to GSSS, although the differences are smaller. We learn that using state-based interference (like the mutex-based by GSSS), but including additional disablers compared to CSS, does mostly not pay off in pruning compared to the pure syntactic interference approximation used by CSS. Finally, at the right, we directly compare CSS against GWSS. We again see that GWSS are orthogonal to CSS (slightly favoring GWSS), thus also confirming our previous theoretical investigation.

We conclude that GWSS are favorable in terms of pruning compared to GSSS, and that the orthogonal pruning power of GWSS and CSS is also reflected in practice. We finally remark that the differences in pruning do not translate to large changes in coverage (i.e., number of solved tasks), which is at most 3. The technical report includes full coverage results.

## 7 Conclusions

We have investigated weak stubborn sets for planning in the light of the operator shifting property. We have learned that sets called weak stubborn sets in earlier work in planning are not weak stubborn sets in the original sense. Based on this finding, we have formally introduced generalized weak stubborn sets for planning, and related them to existing stubborn set approaches.

A promising direction for future research is to further relax the operator shifting property, still yielding safe pruning functions. As we have seen, the more general operator shifting property of CSS compared to stubborn sets is a result of "simply" excluding certain disabling operators. If stubborn sets are applied with syntactic interference, which is often done in practice anyway (e.g., because it allows for a more efficient implementation, see Röger *et al.*, 2020), then the question arises if the operator shifting property can be further relaxed such that larger "operator blocks" are obtained, resulting in smaller sets and higher pruning power. For example, for an operator $o \in T$, can we identify cases where an operator $o'$ can be left out of $T$ even though $o'$ is disabled by $o$? Can such information be synthesized algorithmically and in a generic way? Answers to such questions will potentially show ways to a further level of safe pruning functions.

## Acknowledgments

We thank the anonymous reviewers for their comments, which helped improve the paper. Silvan Sievers has received funding for this work from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 817639). Moreover, Silvan Sievers was partially supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under grant agreement no. 952215.

## References


[Al-Khazraji, 2017] Yusra Al-Khazraji. *Analysis of Partial Order Reduction Techniques for Automated Planning*. PhD thesis, University of Freiburg, 2017.

[Alkhazraji *et al.*, 2012] Yusra Alkhazraji, Martin Wehrle, Robert Mattmüller, and Malte Helmert. A stubborn set algorithm for optimal planning. In *Proc. ECAI 2012*, pages 891–892, 2012.

[Bonet and Geffner, 2001] Blai Bonet and Héctor Geffner. Planning as heuristic search. *AIJ*, 129(1):5–33, 2001.

[Edelkamp, 2001] Stefan Edelkamp. Planning with pattern databases. In *Proc. ECP 2001*, pages 84–90, 2001.

[Gnad *et al.*, 2019] Daniel Gnad, Jörg Hoffmann, and Martin Wehrle. Strong stubborn set pruning for star-topology decoupled state space search. *JAIR*, 65:343–392, 2019.

[Haslum *et al.*, 2007] Patrik Haslum, Adi Botea, Malte Helmert, Blai Bonet, and Sven Koenig. Domain-independent construction of pattern database heuristics for cost-optimal planning. In *Proc. AAAI 2007*, pages 1007–1012, 2007.

[Helmert and Domshlak, 2009] Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proc. ICAPS 2009*, pages 162–169, 2009.

[Helmert, 2006] Malte Helmert. The Fast Downward planning system. *JAIR*, 26:191–246, 2006.

[Helmert, 2009] Malte Helmert. Concise finite-domain representations for PDDL planning tasks. *AIJ*, 173:503–535, 2009.

[Keren *et al.*, 2018] Sarah Keren, Avigdor Gal, and Erez Karpas. Strong stubborn sets for efficient goal recognition design. In *Proc. ICAPS 2018*, pages 141–149, 2018.

[Rintanen, 2008] Jussi Rintanen. Regression for classical and nondeterministic planning. In *Proc. ECAI 2008*, pages 568–572, 2008.

[Röger *et al.*, 2020] Gabriele Röger, Malte Helmert, Jendrik Seipp, and Silvan Sievers. An atom-centric perspective on stubborn sets. In *Proc. SoCS 2020*, pages 57–65, 2020.

[Schulte, 2018] Tim Schulte. Stubborn sets pruning for privacy preserving planning. In *Proc. SoCS 2018*, pages 178–183, 2018.

[Seipp and Helmert, 2018] Jendrik Seipp and Malte Helmert. Counterexample-guided Cartesian abstraction refinement for classical planning. *JAIR*, 62:535–577, 2018.

[Seipp *et al.*, 2017] Jendrik Seipp, Florian Pommerening, Silvan Sievers, and Malte Helmert. Downward Lab. https://doi.org/10.5281/zenodo.790461, 2017.

[Seipp *et al.*, 2020] Jendrik Seipp, Thomas Keller, and Malte Helmert. Saturated cost partitioning for optimal classical planning. *JAIR*, 67:129–167, 2020.

[Sievers and Wehrle, 2021a] Silvan Sievers and Martin Wehrle. Code, benchmarks and experiment data for the IJCAI 2021 paper "On Weak Stubborn Sets in Classical Planning". https://doi.org/10.5281/zenodo.4746377, 2021.

[Sievers and Wehrle, 2021b] Silvan Sievers and Martin Wehrle. On weak stubborn sets in classical planning: Technical report. Technical Report CS-2021-002, University of Basel, Department of Mathematics and Computer Science, 2021.

[Valmari, 1989] Antti Valmari. Stubborn sets for reduced state space generation. In *Proc. APN 1989*, pages 491–515, 1989.

[Wehrle and Helmert, 2012] Martin Wehrle and Malte Helmert. About partial order reduction in planning and computer aided verification. In *Proc. ICAPS 2012*, pages 297–305, 2012.

[Wehrle and Helmert, 2014] Martin Wehrle and Malte Helmert. Efficient stubborn sets: Generalized algorithms and selection strategies. In *Proc. ICAPS 2014*, pages 323–331, 2014.

[Wehrle *et al.*, 2013] Martin Wehrle, Malte Helmert, Yusra Alkhazraji, and Robert Mattmüller. The relative pruning power of strong stubborn sets and expansion core. In *Proc. ICAPS 2013*, pages 251–259, 2013.

[Wilhelm *et al.*, 2018] Anna Wilhelm, Marcel Steinmetz, and Jörg Hoffmann. On stubborn sets and planning with resources. In *Proc. ICAPS 2018*, pages 288–297, 2018.

[Winterer *et al.*, 2017] Dominik Winterer, Yusra Alkhazraji, Michael Katz, and Martin Wehrle. Stubborn sets for fully observable nondeterministic planning. In *Proc. ICAPS 2017*, pages 330–338, 2017.