Stubborn Sets for Fully Observable Nondeterministic Planning

Dominik Winterer, Yusra Alkhazraji
University of Freiburg, Germany
{wintered, alkhazry}@cs.uni-freiburg.de

Michael Katz
IBM Watson Health, Israel
katzm@il.ibm.com

Martin Wehrle
University of Basel, Switzerland
martin.wehrle@unibas.ch

Abstract

Pruning techniques based on strong stubborn sets have recently shown their potential for SAS$^+$ planning as heuristic search. Strong stubborn sets exploit operator independency to safely prune the search space. Like SAS$^+$ planning, fully observable nondeterministic (FOND) planning faces the state explosion problem. However, it is unclear how stubborn set techniques carry over to the nondeterministic setting. In this paper, we introduce stubborn set pruning to FOND planning. We lift the notion of strong stubborn sets and introduce the conceptually more powerful notion of weak stubborn sets to FOND planning. Our experimental analysis shows that weak stubborn sets are beneficial to an LAO* search, and in particular show favorable performance when combined with symmetries and active operator pruning.

Introduction

During the execution of operator sequences, an agent may face situations where she has no control over certain aspects of the world. To handle such situations, it can be helpful to capture the nondeterminism of the world’s dynamic within the domain model. Fully observable nondeterministic planning (FOND) formalism achieves that by allowing operators to have multiple outcomes. That is essentially its key difference from classical planning. Solutions to FOND planning tasks are policies that map states to actions. A policy determines the next action in the final plan, depending on the actual outcome of the previously applied action.

Like in classical planning, a key challenge in FOND planning is the state explosion problem, i.e., the exponential blowup of the state space induced by a compact description of planning tasks. In classical planning, and specifically in SAS$^+$ planning (Bäckström and Nebel 1995), the state explosion problem is often tackled by state pruning techniques (Alkhazraji et al. 2012; Wehrle et al. 2013; Wehrle and Helmert 2014; Domshlak, Katz, and Shleifman 2012; Chen and Yao 2009). Out of these, strong stubborn sets are of a particular interest. Roughly speaking, strong stubborn sets can prune operators in a state if there is an alternative “permutation” plan starting in the same state, and the first operator in the alternative plan is preserved. For example, two operators that work on entirely different state variables can be permuted, and if both are applicable in a state $s$, both application orders in $s$ will result in the same state. Strong stubborn sets will recognize this property of such independent operators, and only allow for applying one of them in $s$, hence pruning states in the induced transition system.

Although SAS$^+$ and FOND planning tasks are syntactically similar, differing only in whether operators may have multiple outcomes, the semantics of these formalisms is different. In particular, “classical” state spaces induced by SAS$^+$ tasks (OR graphs) differ from the more complex state spaces represented by AND/OR graphs induced by FOND planning tasks. This difference poses a challenge for carrying the state pruning techniques from classical planning over to fully observable nondeterministic planning. Recently, symmetry based techniques were adapted in a rather straightforward way to FOND planning (Winterer, Wehrle, and Katz 2016). Strong stubborn sets, however, cannot be carried over to FOND planning in a straightforward way. It is an open question how strong stubborn sets can be lifted to deal with AND/OR graphs, and in particular, how an accurate notion of operator interference for nondeterministic operators would look like.

In this paper we develop the theoretical basis for stubborn sets in FOND planning. We empirically evaluate these pruning techniques, exploiting stubborn sets for pruning LAO$^*$ search in AND/OR graphs. Further, we evaluate whether stubborn sets can contribute on top of other pruning techniques, such as symmetry elimination and active operators (Winterer, Wehrle, and Katz 2016; Wehrle et al. 2013). In more detail, the paper provides the following contributions.

- We generalize the theory of stubborn set pruning from SAS$^+$ planning to FOND planning. The generalization is based on a corresponding notion of interference.

- We introduce weak stubborn sets for FOND planning. Like strong stubborn sets, weak stubborn sets (Valmari 1989) have been originally introduced for model checking Petri nets (Petri 1962). Weak stubborn sets potentially allow for more pruning than strong stubborn sets, but require a more involved description for the original formalism used by Valmari (1989). Unlike strong stubborn sets, weak stubborn sets have not been investigated for planning before. We also introduce a nondeterministic variant of strong stubborn sets for FOND planning.
• We empirically evaluate the practical potential of stub-
born set pruning to FOND planning with LAO*. In par-
icular, inspired by recent work for classical planning that
combines strong stubborn sets and symmetry elimina-
tion (Wehrle et al. 2015), we combine weak stubborn sets with
structural symmetries for FOND tasks (Winterer, Wehrle,
and Katz 2016) and with active operators pruning (Chen
and Yao 2009; Wehrle et al. 2013). Our empirical inves-
tigation shows that the combination of all these pruning
techniques yields promising performance gains on top of
the respective baselines.

Preliminaries
We use an SAS+ based notation to model fully observ-
able nondeterministic planning tasks with a finite set of finite-
domain state variables \(V\). Every variable \(v\) in \(V\) has a finite
domain \(\text{dom}(v)\). An assignment \(v = d\) for \(v \in V\) and \(d \in \text{dom}(v)\) is called a fact. A partial state \(s\) is a function from
variables \(\text{vars}(s) \subseteq V\) to values in the domains of \(\text{vars}(s)\), whereas all variables in \(V \setminus \text{vars}(s)\) have an undefined value \(\perp\). We denote the value of \(s[v]\) in \(s\) with \(s[v]\) (including \(s[v] = \perp\) if \(v \in V \setminus \text{vars}(s)\)). A state is a partial state where all
values are defined, i.e., with \(\text{vars}(s) = V\). We consider
fully observable nondeterministic (FOND) planning tasks.
As a key difference to classical planning, operators in FOND
planning tasks can have several effects. Only one of these
effects is applied when the operator is executed. Formally,
FOND planning tasks are defined as follows.

**Definition 1** (fully observable nondeterministic planning
task). A fully observable nondeterministic (FOND) plann-
ing task is a tuple \(\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_*)\), where
• \(\mathcal{V}\) is a finite set of finite-domain state variables,
• \(\mathcal{O}\) is a finite set of nondeterministic operators,
• \(s_0\) is the initial state,
• \(s_*\) is a partial state, called goal state.

A nondeterministic operator \(o \in \mathcal{O}\) has the form \((\text{pre}(o) \mid \text{effs}(o))\), where \(\text{pre}(o)\) is a partial state that denotes the pre-
condition of \(o\) and \(\text{effs}(o)\) is the set of effects of \(o\), where
each effect \(\text{eff}(o) \in \text{effs}(o)\) is a partial state. The degree of
an operator \(o\), denoted as \(\text{deg}(o)\), is the number of its non-
deterministic effects. Deterministic operators have degree of
1, i.e., \(\text{effs}(o) = \{\text{eff}^1(o)\}\). For simplicity, we write \(\text{eff}(o)\)
instead of \(\text{eff}^1(o)\) for deterministic operators. Throughout
this paper, we assume unit costs for all operators.
In the following, we will denote a nondeterministic op-
erator as an operator unless stated otherwise. The all-
outcome determination of nondeterministic operator \(o\) is
\(o^{[1]}, \ldots, o^{[k]} (k = \text{deg}(o))\) where \(o^{[i]} = \langle \text{pre}(o), \{\text{eff}^i(o)\}\rangle\)
for \(1 \leq i \leq k\) is called an outcome of \(o\). We use \(\mathcal{O}_{\text{set}}\) to
denote the set of all outcomes of operators in \(\mathcal{O}\). An
operator \(o\) is applicable in a state \(s\) if \(s[v] = \text{pre}(o)[v]\)
holds for all \(v \in \text{vars}(\text{pre}(o))\). If \(o\) has an empty pre-
condition (i.e., it has no precondition), then \(o\) is applicable
in every state. The empty precondition is denoted by \(\perp\).
If operator \(o\) is applicable in \(s\), the set of successor states
\(\text{o}(s) := \{o^{[i]}(s) \mid i \in \{1, \ldots, \text{deg}(o)\}\}\) of \(s\) is obtained
from \(s\) by applying the outcomes \(o^{[i]}(s)\) in \(s\), where the state
\(o^{[i]}(s)\) is obtained from \(s\) by setting the values of variables
in \(\text{vars}(\text{eff}^i(o))\) to their values in \(\text{eff}^i(o)\) and leaving
the remaining variable values unchanged. For simplicity of no-
tation: If \(o(s)\) includes a single state, we refer to it as a state
rather than a set of states. We denote the set of applicable
operators in \(s\) by \(\text{app}(s)\).

Solutions to FOND planning tasks are policies subsuming
the plan notion of plans in classical planning (Cimatti et al.
2003). Formally, a policy \(\pi\) is a mapping \(\pi : S \mapsto O \cup \{\perp\}\),
which maps states to operators or \(\pi\) is undefined (i.e. \(\pi(s) = \perp\)).
We denote the sequential application of the operators
determined by \(\pi\) as following \(\pi\). A policy \(\pi\) is called weak
if \(\pi\) defines at least one path from the initial state to a goal
state when following \(\pi\). In this case, \(\pi\) is called a weak plan
for \(\Pi\). A policy \(\pi\) is closed if following \(\pi\) either leads to a
goal state, or to a state where the policy is defined. It is
proper if from every state visited following \(\pi\), there exists a
path to a goal state following \(\pi\). A policy that is closed and
proper is called a strong cyclic plan for \(\Pi\), the plan notion of
interest thorough this paper. Furthermore, \(\pi\) is acyclic if
it does not revisit already visited states. A closed and proper
acyclic policy is called a strong plan for \(\Pi\).
Informally, a weak plan is a sequence of operators which
leads to the goal if all nondeterministic operators’ outcomes
were deterministic, which corresponds to the plan notion
in classical planning. A strong plan guarantees that a goal
state is reached, and an upper bound on the number of plan
steps exists. In contrast to strong plans, strong cyclic plans
reach a goal state after a number of steps, but no such upper
bound on the plan steps can be provided a priori. However,
strong cyclic planning is based on the fairness assumption,
i.e., there is a nonzero probability that a goal state can be
reached when following a strong cyclic plan.

**Stubborn Sets for Classical Planning**
The stubborn set method was originally introduced in
computer-aided verification and used for dead end detection
in Petri nets (Valmari 1989). Stubborn sets have been
investigated and evaluated for optimal planning in the work
of Alkhazrajji et al. (2012). In a nutshell, a stubborn set for
some state \(s\) is a sufficient subset of operators such that re-
stricting the successor generation to the applicable operators
within this stubborn set preserves a permutation of each plan
from \(s\). In this section, we will introduce all necessary pre-
liminaries for stubborn sets in classical planning.

**Definition 2** (disabling, conflicting, strong and weak inter-
ference). Let \(o_1\) and \(o_2\) be deterministic operators.
• \(o_1\) disables \(o_2\) if there exists some \(v \in \text{vars}(\text{eff}(o_1)) \cap \text{vars}(\text{pre}(o_2))\) with \(\text{eff}(o_1)[v] \neq \text{pre}(o_2)[v]\).
• \(o_1\) and \(o_2\) conflict if there exists some \(v \in \text{vars}(\text{eff}(o_1)) \cap \text{vars}(\text{eff}(o_2))\) with \(\text{eff}(o_1)[v] \neq \text{eff}(o_2)[v]\).
• \(o_1\) and \(o_2\) strongly interfere if \(o_1\) disables \(o_2\), \(o_2\) disables
\(o_1\), or \(o_1\) and \(o_2\) conflict.
• \(o_1\) weakly interferes with \(o_2\) if \(o_1\) disables \(o_2\), or \(o_1\) and
\(o_2\) conflict.
In contrast to strong interference, weak interference is not a symmetric relation. In other words, if operator \( o_1 \) weakly interferes with operator \( o_2 \), it does not imply that \( o_2 \) weakly interferes with \( o_1 \). The two interference definitions are approximations in the sense that they are state-independent whereas the true operator dependencies depend on the current state. There are two further ingredients for the stubborn sets. Let \( s \) be a state. A **disjunctive action landmark** for a partial state \( s' \) in \( s \) is a set of operators with the property that at least one operator from this set occurs in every operator sequence that leads from \( s \) to any state that satisfies \( s' \). A **necessary enabling set** for an operator \( o \) in \( s \) is a disjunctive action landmark for \( \text{pre}(o) \) in \( s \).

Two variants of stubborn sets are known in the literature: **strong** and **weak** stubborn sets. Their sole difference is the interference relation. In planning, however, only the strong variant of stubborn sets has been investigated so far. In our work, we will also investigate weak stubborn sets. Based on Definition 2, the notion of weak stubborn sets for classical planning boils down to the following definition.

**Definition 3** (weak stubborn sets). Let \( \Pi = (\mathcal{V}, \mathcal{O}, s_0, s_*) \) be a classical planning task and \( s \) be a state. A set \( T_s \subseteq \mathcal{O} \) is a **weak stubborn set** (WSS) in \( s \) if:

1. \( T_s \) contains a disjunctive action landmark for \( s_* \) in \( s \).
2. For each operator \( o \in T_s \) with \( o \notin \text{app}(s) \), \( T_s \) contains a necessary enabling set for \( o \) in \( s \).
3. For each operator \( o \in T_s \) with \( o \in \text{app}(s) \), \( T_s \) contains all operators \( o' \) with which \( o \) weakly interferes.

If we exchange weak interference by strong interference, we end up with a **strong** stubborn set. Obviously, weak stubborn sets dominate strong stubborn sets in terms of pruning power, in the sense that every strong stubborn set is a weak stubborn set, but not vice versa. The proof of retaining completeness and optimality given by Alkhazraji et al. (2012) directly generalizes to weak interference. We emphasize that there is a difference in weak and strong interference computation. While strong interference might be refined by mutex reasoning (i.e., two operators \( o_1 \) and \( o_2 \) cannot strongly interfere if there is a variable \( v \in \text{vars}(\text{pre}(o_1)) \cap \text{vars}(\text{pre}(o_2)) \) and \( \text{pre}(o_1)(v) \neq \text{pre}(o_2)(v) \)), the mutex reasoning violates the completeness of weak stubborn sets. We refer to our technical report for details.

**Stubborn Sets for FOND**

For classical planning tasks, the solutions are consecutive operator applications leading from the initial state to one of the goal states. Hence, for an operator sequence \( o_1, \ldots, o_n \) of pairwise independent operators, all permutations of \( o_1, \ldots, o_n \) lead to the same state. This property is the core idea of stubborn sets. For FOND planning tasks, however, solutions cannot be represented as consecutive operator sequences and it is unclear how strong cyclic plans permute. As a first idea to face this theoretical problem, we might determine the nondeterministic operators and use the stubborn set theory from classical planning. More precisely: For a given nondeterministic planning problem \( \Pi \), a straightforward approach would be to directly apply the original definition of strong stubborn sets on the all-outcome determinization of \( \Pi \), and additionally, to add for every outcome \( o^{(i)} \) of a nondeterministic operator \( o \) every other outcome of \( o \) to the candidate stubborn set, in order to respect \( o \)'s nondeterministic nature. However, as the following example shows, such an approach is incomplete.

**Example 1.** Consider the following all-outcome determinization \( \Pi_{\text{det}} = (\mathcal{V}, \mathcal{O}_{\text{det}}, s_0, s_*) \) of nondeterministic planning task \( \Pi \) with variables \( \mathcal{V} = \{ v_1, v_2 \} \) and the following operators:

- \( o_{11} = \{ v_1 = 1, v_2 = 1 \} \)
- \( o_{12} = \{ v_1 = 1, v_2 = 2 \} \)
- \( o_{21} = \{ v_1 = 2, v_2 = 3 \} \)
- \( o_{22} = \{ v_1 = 2, v_2 = 4 \} \)
- \( o_{14} = \{ v_1 = 2, v_2 = 5 \} \)

The initial state is \( s_0 = \{ v_1 = 0, v_2 = 0 \} \), and the goal is \( s_* = \{ v_2 = 5 \} \). The set \( \{ o_{11}, o_{12}, o_{21}, o_{22}, o_{23}, o_{24}, o_{25}, o_{26} \} \) is a disjunctive action landmark in \( s_0 \) which we add to the candidate stubborn set \( T_{s_0} \). As all operators in this set are inapplicable in \( s_0 \), we have to add a necessary enabling set for all of them. A valid choice for these necessary enabling sets is based on selecting the unsatisfied conditions \( v_2 = 1, v_2 = 2, v_2 = 3 \) and \( v_2 = 4 \) in the preconditions of \( o_{11}, o_{12}, o_{21}, o_{22}, \) respectively, and to add the determined operators that set these conditions to true. These achieving operators correspond to all outcomes of \( o_2 \) and \( o_3 \), which are applicable in \( s_0 \) but not weakly interfering with any operator not in \( T_{s_0} \). Hence, we finally get \( T_{s_0} = \{ o_{11}, o_{12}, o_{21}, o_{22}, o_{14}, o_{25}, o_{26} \} \). However, \( T_{s_0} \) is insufficient for our purpose because every strong plan from \( s_0 \) has to start with \( o_{11} \). Depending on the nondeterministic outcome of \( o_1 (v_1 = 1 \text{ or } v_1 = 2) \), \( o_2 \) or \( o_3 \) can be applied to satisfy the precondition of an operator to reach the goal. In contrast, starting with \( o_2 \) and applying \( o_1 \) afterwards might lead to outcomes where no goal is reachable any more (e.g., \( v_1 = 2 \) and \( v_2 = 2 \)). The analogous situation occurs when starting with \( o_3 \) and applying \( o_1 \) afterwards (Figure 1).

![Figure 1](image-url)
We observe: Weak interference in the all-outcome determination is an insufficient criterion for the coupling behaviour of nondeterministic operators and the stubborn sets cannot be carried over to FOND planning in a straightforward way. Hence, other ideas are needed. We follow a stepwise approach. First, we give a new criterion based on deterministic operators. Then we prove a syntactic version of the criterion. Finally, we adapt the criterion to the FOND setting and use it as a basis for our pruning technique.

We introduce the idea of an operator being attached to the front of a weak plan in a given state such that the resulting sequence remains a weak plan from that state (Figure 2). Formally, the following definition reflects this idea.

**Definition 4** (attachable operator). Let \( o_1 \) and \( o_2 \) be operators in \( O_{\det} \), \( \pi \) be an operator sequence, and \( s \) be a state. Operator \( o_1 \) is attachable to \( o_2 \) in state \( s \) if for every weak plan \( o_2 \pi \) from \( s \), \( o_1 o_2 \pi \) is a weak plan from \( s \).

Checking the attachability property is computationally intractable. To be precise, this property is state-dependent, i.e., it needs to be checked in each visited state. Moreover, verifying this property amounts to finding two weak plans for \( s \), one that starts with \( o_2 \) and another one that starts with \( o_1 \). Therefore, it is necessary to come up with an approximation based on a syntactic check of operators which we formulate using two new definitions. Let \( \Pi = (\mathcal{V}, O, s_0, s_\pi) \) be a FOND planning task and \( o_1 \) be an operator in \( O_{\det} \). The **disabled set** \( \text{dis}(o_1) \) is defined as the set of operator-variable pairs \((o_2, v) \in O_{\det} \times \mathcal{V} \) such that \( o_1 \) disables \( o_2 \) on \( v \), i.e., \( v \in \text{vars}(\text{eff}(o_1)) \cap \text{vars}(\text{pre}(o_2)) \) and \( \text{eff}(o_1)[v] \neq \text{pre}(o_2)[v] \). The **negated goals set** \( \text{neg}(o_1) \) is defined as the set of goal variables with which \( o_1 \) conflicts, i.e., \( \text{eff}(o_1)[v] \neq \text{neg}(o_2)[v] \).

**Definition 5** (syntactically attachable operator). Let \( o_1 \) and \( o_2 \) be two operators in \( O_{\det} \). We say that \( o_1 \) is syntactically attachable to \( o_2 \) if

1. \( o_1 \) does not disable \( o_2 \).
2. \( \text{dis}(o_1) \subseteq \text{dis}(o_2) \), and
3. \( \text{neg}(o_1) \subseteq \text{neg}(o_2) \).

The syntactic attachability property is state-independent and does not suffice on its own to verify attachability since it states nothing about the applicability of operators. For instance, \( o_1 \) can be inapplicable in state \( s \) in which \( o_2 \) is applicable, while all the conditions in Definition 5 hold. Therefore, stipulating that \( o_1 \) is applicable in \( s \), in addition to being syntactically attachable to \( o_2 \), is essential for implying attachability.

**Theorem 1.** Let \( \Pi = (\mathcal{V}, O, s_0, s_\pi) \) be a FOND planning task, \( s \) be a state of \( \Pi \), and \( o_1 \) and \( o_2 \) be operators in \( O_{\det} \cap \text{app}(s) \). If \( o_1 \) is syntactically attachable to \( o_2 \), then \( o_1 \) is attachable to \( o_2 \) in \( s \).

For the proof of Theorem 1 we again refer to the technical report. In the following section we emphasize the difference between weak interference and attachability.

**Weak Interference vs. Attachability.** According to weak interference, we obtain equivalent weak plans by permuting weakly non-interfering operators. On the other hand, attachability allows introducing new operators to the beginning of weak plans such that the resulting sequence is a weak plan, too. As a first step towards a stubborn set formalism for FOND planning, we clarify the relationship between attachability and weak interference. At first sight, it might seem that attachability subsumes weak interference. However, as it turns out, both notions are incomparable, i.e., if operator \( o_1 \) is syntactically attachable to operator \( o_2 \), this does not imply that \( o_1 \) does not weakly interfere with \( o_2 \).

**Example 2.** Consider a FOND planning task \( \Pi \) with variables \( \mathcal{V} = \{v_1, v_2, v_3, v_4\} \) and the following operators:

- \( o_1 = \langle 1 | \{v_2 := 1, v_4 := 1\} \rangle \)
- \( o_2 = \langle 3 | \{v_3 := 1, v_4 := 2\} \rangle \)

The initial state is \( s_0 = \{v_1 = 0, v_2 = 0, v_3 = 0, v_4 = 0\} \), the goal partial state \( s_g = \{v_2 = 1, v_3 = 1\} \). We observe that operators \( o_1 \) and \( o_2 \) are mutually syntactically attachable, i.e., \( \text{dis}(o_1) \subseteq \text{dis}(o_2) = \emptyset \), and \( \text{neg}(o_1) \subseteq \text{neg}(o_2) = \emptyset \). However, they also mutually weakly interfere because they conflict on variable \( v_4 \). This fundamental difference of weak interference and attachability gives rise to a new concept of state space pruning. We obtain a stubborn set based solely on attachability by replacing rule 3 in Definition 3 with the following rule:

3'. For each operator \( o \in T_s \) with \( o \in \text{app}(s) \), \( T_s \) contains all operators \( o' \) to which \( o \) is not syntactically attachable.

This yields a novel notion of attachability-based stubborn sets which is a completeness-preserving pruning technique. However, due to the nature of attachability, it can lead to suboptimal weak plans. We further show, by the following example, that stubborn sets based on weak interference and stubborn sets based on attachability are two different pruning techniques such that none of them dominates the other with respect to pruning power.

**Example 3.** Consider a FOND planning task \( \Pi \) with variables \( \mathcal{V} = \{v_1, v_2, v_3\} \) and the following operators:

- \( o_1 = \langle \top | \{v_1 := 2, v_2 := 1\} \rangle \)
- \( o_2 = \langle \top | \{v_1 := 1\} \rangle \)
- \( o_3 = \langle v_1 := 1, v_2 := 1 | \{v_3 := 1\} \rangle \)

The initial state is \( s_0 = \{v_1 = 0, v_2 = 0, v_3 = 0\} \) and the goal state is \( s_g = \{v_1 = 1, v_2 = 1, v_3 = 1\} \). Let \( T_s \) denote a stubborn set that uses only syntactic attachability. Since

\[\text{We do not provide a formal proof, as it is an intermediate step towards our contribution. However, it is easy to see that attachability is a sufficient condition for completeness: } \text{dis}(o_1) \subseteq \text{dis}(o_2) \text{ guarantees that } v_4 \text{ does not occur in some operator’s precondition s.t. } o_2 \text{ enables that operator w.r.t. } v_4 \text{ and } o_1 \text{ disables it. Hence, conflicting on } v_4 \text{ is not critical in this case for preserving plans.}\]
\{o_2\} is a disjunctive action landmark for \(s_1\) in \(s_0\), we can use it to initialize the stubborn set: \(T_a(s_0) = \{o_2\}\). Because \(o_2 \in \text{app}(s_0)\), we need to add the operators to which \(o_2\) is not attachable. Since \(\text{dis}(o_2) = \emptyset\) and \(\text{neg}(o_2) = \emptyset\), \(o_2\) is attachable to both \(o_1\) and \(o_3\). This means that the computation will terminate on \(T_a(s_0) = \{o_2\}\). On the other hand, if we compute \(T_s\), the stubborn set based on weak interference, the result is different. We initialize \(T_s(s_0)\) with \(o_2\) as before. \(o_2\) is applicable, which means we need to add operators with which \(o_2\) weakly interferes. Obviously, \(o_2\) weakly interferes with \(o_1\) because they conflict on variable \(v_1\). Therefore, \(o_1\) is added to \(T_s(s_0)\). Also, \(o_1\) is applicable and it disables \(o_3\), hence \(o_3\) is added to \(T_s(s_0)\), which means all operators are applied in \(s_0\) \(T_s(s_0) = \{o_1, o_2, o_3\}\).

Furthermore, each method can result in a different plan: depending on the search algorithm, using only syntactic attachability might result in plan \(o_2 o_1 o_2 o_3\), while using only weak interference might lead to \(o_1 o_2 o_3\), which is optimal.

**Example 4.** Consider a FOND planning task \(\Pi\) with variables \(V = \{v_1, v_2\}\) and the following operators:

- \(o_1 = \langle T \mid \{v_1 := 1\}\rangle\)
- \(o_2 = \langle T \mid \{v_2 := 1\}\rangle\)
- \(o_3 = \langle v_1 = 0 \mid \{v_2 := 1\}\rangle\)

The initial state is \(s_0 = \{v_1 = 0, v_2 = 0\}\) and the goal state \(s_1 = \{v_1 = 1, v_2 = 1\}\). We compute \(T_a(s_0)\): The set \(\{o_1\}\) is a disjunctive action landmark in \(s_0\), so we add it to \(T_a(s_0)\). Since \(o_1\) is applicable, we add operators to which it is not attachable. Because \(\text{dis}(o_1) = \{o_1\} \not\subseteq \text{dis}(o_2) = \emptyset\), \(o_1\) is not attachable to \(o_2\) and therefore \(o_2\) is added to \(T_a(s_0)\). In addition, \(o_3\) is added because \(o_1\) disables \(o_3\) and hence not attachable to it, ending in \(T_a(s_0) = \{o_1, o_2, o_3\}\). Using weak interference: we add \(o_1\) to \(T_a(s_0)\) as a disjunctive action landmark. It is applicable so we need to add the operators with which \(o_1\) weakly interferes. We add \(o_3\) only because \(o_1\) disables \(o_3\), but we do not add \(o_2\) because neither \(o_1\) nor \(o_3\) weakly interferes with \(o_2\). Therefore, \(T_s(s_0) = \{o_1, o_3\}\). We conclude this section with the following corollary, summarizing our observations.

**Corollary 1.** Stubborn sets based on weak interference and stubborn sets based on attachability are incomparable in terms of pruning power.

**Nondeterministic Weak Stubborn Sets**

While attachability is a sufficient criterion for stubborn sets pruning, it could lead to unnecessary state explorations (e.g. see plan length in Example 3). We propose a stubborn set method based on a combination of attachability and weak interference for FOND planning. The definition of a disjunctive action landmark can be extended to FOND planning in a straightforward way. In the context of FOND planning, a **disjunctive action landmark** for a partial state \(s'\) in a state \(s\) is a set of nondeterministic operators with the property that some outcome of an operator from this set occurs on every path from \(s\) to any state that satisfies \(s'\). A necessary enabling set for an operator \(o\) in \(s\) is a disjunctive action landmark for \(\text{pre}(o)\) in \(s\). The following definitions extend syntactic attachability and weak interference to nondeterministic operators. Let \(o_1\) and \(o_2\) be two operators in \(\mathcal{O}\). We say that \(o_1\) **accords** with \(o_2\) if outcome \(o_1[i]\) is syntactically attachable to outcome \(o_2[i]\) for all \(i \in \{1, \ldots, \text{deg}(o_1)\}\) and \(j \in \{1, \ldots, \text{deg}(o_2)\}\).

**Definition 6** (nondeterministic weak stubborn set). Let \(\Pi = (\mathcal{V}, \mathcal{O}, s_0, s_1)\) be a nondeterministic planning task, and \(s\) be a state. A set \(T_s \subseteq \mathcal{O}\), is a **nondeterministic weak stubborn set** (NWSS) in \(s\) if the following conditions hold:

1. \(T_s\) contains a disjunctive action landmark for \(s_1\) in \(s\).
2. For each operator \(o \in T_s\) with \(o \notin \text{app}(s)\), \(T_s\) contains a necessary enabling set for \(o\) in \(s\).
3. For each operator \(o \in T_s\) with \(o \in \text{app}(s)\), \(T_s\) contains all nondeterministic operators \(o'\) (\(\text{deg}(o') > 1\)) with which \(o\) does not accord.
4. For each operator \(o \in T_s\) with \(o \in \text{app}(s)\), \(T_s\) contains all operators with which \(o\) weakly interferes.

Nondeterministic weak stubborn sets can be seen as hybrid notion combining attachability with weak interference.
Points (1) and (2) of Definition 6 are the same in Definition 3. Accordance, in point (3), is a necessary property for changing the application order of nondeterministic operators in AND/OR graphs. However, in some cases, pruning can be performed by merely permuting operator sequence (e.g. if the involved operators are deterministic) without the need to attach extra operators, which cannot be guaranteed by accordance alone. For this reason, we have chosen to make the distinction between deterministic and operators with degree at least two and use only weak interference when processing deterministic operators (point 4), i.e., if a planning task has only deterministic operators, then only points (1), (2), and (4) of Definition are relevant. In the following theorem we state the main result of this paper.

**Theorem 2.** Restricting the successor generation to an NWSS in every state is completeness-preserving for strong cyclic planning.

*Proof.* Let s be a state from which a strong cyclic plan exists and let π be one such plan. Let $T_s$ be a nondeterministic weak stubborn set for s. We show that either (i) the operator $\pi(s)$ is in $T_s$, or (ii) there exists $o \in T_s$ such that $o \in \text{app}(s)$ and there exists a strong cyclic plan from each $s' \in o(s)$.

Let $\pi' = o_1 \ldots o_n$ be any sequence of operators from $\pi$ that defines an acyclic weak plan from s. $T_s$ contains a disjunctive action landmark for $s_*$ in s and thus it contains an operator from $\pi'$. Let $o_i$ be such an operator with smallest index. Then $o_i \in \text{app}(s)$ (otherwise its necessary enabling set would be contained in $T_s$, mandating an operator from the necessary enabling set to appear on $\pi'$ before $o_i$). If $i = 1$, we are done. Otherwise, consider two cases. If the operator $o_i$ is deterministic, we see that $o_i^{[1]} 0_1 \ldots 0_{i-1} 0_{i+1} o_n$ is a weak plan from s and thus a strong cyclic plan. If $o_i$ is a strictly nondeterministic operator, for every outcome $o_i^{[j]}$, $o_i^{[j]} \pi'$ is a weak plan as $o_i^{[j]}$ accords with every operator of smaller index within $\pi'$. We thus obtain a strong cyclic plan by first applying $o_i$ in s, followed by applying $\pi'$ in every possible outcome in $o_i(s)$.

We also provide a nondeterministic strong variant of stubborn set (NSSS) by modifying Definition 6 as follows: In point 3, all operators that are do not mutually accord with o are added to the stubborn set. In point 4, all operators that strongly interfere with o are added to the stubborn set.

**Corollary 2.** NSSS inherit the completeness property from NWSS.

*Proof.* The definition of NSSS subsumes the definition of NWSS; hence, the proof of Theorem 2 shows also that NSSS is a completeness-preserving pruning technique.

**Experimental Evaluation**

In this section, we empirically investigate the potential of nondeterministic stubborn sets when applied as the only pruning technique or on top of other pruning techniques such as structural symmetries and active operators pruning. For our investigation, we have implemented both strong and weak stubborn set variants, as well as active operators on top of an adaptation of Fast Downward (Helmert 2006) to FOND planning (Winterer, Wehrle, and Katz 2016), which already included symmetry based pruning. Our code is available upon request.

As baseline for our investigations, we employ LAO* using FF heuristic (Hoffmann and Nebel 2001) (based on all-outcome determinization of FOND planning tasks). All configurations are built on top of this baseline. For all stubborn sets approaches, the disabling relation and achievers were entirely precomputed, while the interference relation was computed during the search and then cached for later use. The benchmark set consists of all IPC-08 FOND domains, scaled-up versions of these domains by Christian Muise and other FOND domains commonly used in the literature. All our experiments were conducted on a cluster equipped with Intel Xeon E5-2650 v2 CPUs running at 2.6 GHz. For each run, the time limit and memory bound were 30 minutes and 2 GB respectively.

**Stubborn Sets for FOND**

First of all, we want to find out whether nondeterministic strong and nondeterministic weak stubborn sets lead to noticeable performance improvement. To that end, we com-
The configurations applying pruning with each of the stubborn sets on top of the baseline (FF). The per domain coverage is depicted in Table 1 and per instance generated nodes are shown in Figure 3, pairwise comparing the three configurations. The configurations applying nondeterministic strong and weak stubborn sets are denoted by NSSS and NWSS, respectively. First, note that NSSS performs similarly to FF, loosing one task each in three domains, and increasing coverage by two tasks in one domain. In total, coverage is reduced by one instance which goes along with only slight reductions in node generations: In Figure 3a, most of the tasks appear on or near the diagonal.

Comparing NWSS to FF, it looses one task each in two domains and gains five tasks in one domain and one task each in three domains, overall increasing coverage by six tasks. Looking at Figure 3b, indeed pruning helps in most cases to reduce the number of generated nodes, in a number of cases the decrease almost reaching two orders of magnitude. Note that there is an exponential reduction in generated nodes on a part of the FIRST-RESPONDERS domain. Figure 3c shows that this gain carries over to the comparison with NSSS, reflecting the theoretical dominance of nondeterministic weak stubborn sets over the strong variant. It should be pointed out that due to the non-optimal nature of the LAO* algorithm, the theoretical dominance does not necessarily translate to a dominance in the number of generated nodes. To summarize, nondeterministic weak stubborn are beneficial to an LAO* search whereas nondeterministic strong stubborn sets are beneficial to a lesser extent.

Table 1: Coverage of baseline (FF), strong stubborn sets (NSSS) and weak stubborn sets (NWSS) and the state of the art FOND planner (PRP).

<table>
<thead>
<tr>
<th>Coverage</th>
<th>FF</th>
<th>NSSS</th>
<th>NWSS</th>
<th>PRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCKSWORLD (30)</td>
<td>22</td>
<td>21</td>
<td>21</td>
<td>30</td>
</tr>
<tr>
<td>CHAIN-OF-ROOMS-FIXED (10)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>EARTH-OBSERVATION (40)</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>40</td>
</tr>
<tr>
<td>FAULTS (55)</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>FIRST-RESPONDERS (100)</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>100</td>
</tr>
<tr>
<td>FOREST (50)</td>
<td>8</td>
<td>10</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>BLOCKSWORLD-NEW (50)</td>
<td>16</td>
<td>15</td>
<td>15</td>
<td>37</td>
</tr>
<tr>
<td>ELEVATORS (15)</td>
<td>14</td>
<td>14</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>EX-BLOCKSWORLD (15)</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>FAULTS-NEW (190)</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>FIRST-RESPONDERS-NEW (95)</td>
<td>30</td>
<td>30</td>
<td>31</td>
<td>94</td>
</tr>
<tr>
<td>FOREST-NEW (90)</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>88</td>
</tr>
<tr>
<td>TIDYUP-MDP (10)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>TIREWORLD (15)</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>TRIANGLE-TIREWORLD (40)</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>40</td>
</tr>
</tbody>
</table>

Sum (645) 523 522 529 808

Figure 5: Total time for NWSS vs. the baseline (FF).

Combination with other Pruning Techniques

Recently, it has been shown that structural symmetries and stubborn sets can be successfully combined for classical planning (Wehrle et al. 2015). Hence, a natural question arises: Do we get similar results when combining weak stubborn sets with structural symmetries for FOND planning? And more generally, to which extent can we observe synergy effects when combining pruning techniques for FOND planning? To take a step towards these questions, we investigate combinations of nondeterministic weak stubborn sets with the following two pruning techniques.

- **Symmetry Elimination:** Symmetry elimination considers equivalence classes of symmetrical states and allows for using representative states of each equivalence class. Many approaches have shown their potential in several contexts in classical planning. We use the variant of structural symmetries for FOND planning, recently proposed by Winterer, Wehrle, and Katz (2016).

- **Active operators pruning:** The pruning technique was introduced by Chen and Yao (2009) and further investigated by Wehrle et al. (2013). In a nutshell, given a state s, an operator o is considered active if there exists a weak plan from s starting with o. A sufficient criterion can be formulated based on domain transition graph. The successor generation in a state s is restricted to active operators.

We remark that combining these pruning techniques in a straightforward fashion results in a combined pruning technique that is safe. For our investigation, we consider all possible combinations of symmetry elimination and active operators pruning with and without nondeterministic weak stubborn sets. We report the overall coverage results in Table 2. Going beyond the overall coverage, Figure 4 depicts the generated nodes comparison. We observe that all non-combined pruning techniques improve coverage. Further, nondeterministic weak stubborn sets is the most beneficial configuration out of those (529 solved instances), followed by symmetry elimination (526 solved instances) and active operator pruning (524 solved instances). Especially, the configuration with additional active operator pruning leads to a...
Table 2: Coverage results for all combinations of NWSS, SYM and active operator pruning (all ops vs active ops).

<table>
<thead>
<tr>
<th>Pruning Technique</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all ops</td>
</tr>
<tr>
<td>FF</td>
<td>523</td>
</tr>
<tr>
<td>NWSS</td>
<td>529</td>
</tr>
<tr>
<td>SYM</td>
<td>526</td>
</tr>
<tr>
<td>NWSS + SYM</td>
<td>532</td>
</tr>
</tbody>
</table>

substantial coverage increase of 15 additionally solved instances (compared to the baseline FF) which reflects the fact that this configuration is also clearly generating the least nodes (see Figure 4). As another insight, we see that applying NWSS on top of symmetry reduction (SYM) leads to a clear performance increase, both with and without additional active operators pruning. Evaluating the gain from applying active operators pruning, this pruning technique is only moderately beneficial both as a single pruning technique and when combined with symmetry reduction and nondeterministic stubborn sets.

To investigate the anytime performance of our approaches, we provide a plot in Figure 6, reflecting for each time point the number of tasks solved by each of the configurations up to that time. Observe that the combination of all three methods dominates all other configuration. Also it can be seen that combining nondeterministic weak stubborn sets with symmetry elimination improves the runtime behavior, which makes them worth using in a practical setting.

Related Work

Pruning techniques in classical planning are directly related to this work. Beside stubborn sets, there is a variety of partial order reduction methods that have been applied to planning (Coles and Coles 2010; Nissim, Apsel, and BRAFMAN 2012; CHEN and YAO 2009). Moreover, symmetries have been investigated for classical planning as well (FOX and LONG 1999; 2002; RINTANEN 2003; COLES and COLES 2010; POCHer, ZOHAR, and Rosenschein 2011; DOMshLak, Katz, and Shleyfman 2012; 2013; Shleyfman et al. 2015).

Our work is also related to various approaches in FOND planning. The LAO∗algorithm was also used for FOND planning by MATTmüller et al. (2010) and was originally introduced in the context of MDPs (HANSEN and ZILBERstein 2001). As a key difference to the LAO∗algorithm, determinization-based approaches work on the all-outcome determinization of FOND planning tasks. The current state of the art is represented by determinization-based planner Planner for Relevant Policy (PRP) by MUISE, McIlraith, and BECK (2012). PRP uses multiple runs of a classical planner and regression to find a strong cyclic plan for a given FOND problem. Another recent regression-based FOND planner is GRENDél (RAMírez and Sardiña 2014). Unlike PRP, GRENDél does not rely on determinization but combines regression with a symbolic fixed-point computation. In addition, approaches that generalize classical planning concepts to FOND planning are particularly related to our work. For example, conditional effects for FOND planning (MUise, McILraith, and Belle 2014), or pattern databases (MATTmüller et al. 2010).

Conclusion

We provided the theoretical basis for stubborn sets pruning in FOND planning by introducing two stubborn set variants for FOND: nondeterministic weak and strong stubborn sets. We empirically showed that nondeterministic weak stubborn sets are beneficial to an LAO∗ search. Moreover, we showed that the dominance of nondeterministic weak stubborn sets over the strong variant carries over in practice. We also investigated the combination of nondeterministic weak stubborn sets with symmetry elimination and active operators pruning. The empirical results show that combining all three pruning methods leads to the strongest configuration. For future work, encouraged by this insight, it is particularly interesting to investigate the potential of combining other pruning techniques in the context of classical and FOND planning. Moreover, as pruning techniques may slow down the search on certain instances, it is beneficial to design algorithms that can detect (prior to search) whether or not a particular pruning technique should be used during search.

Acknowledgements

We thank the anonymous reviewers for their keen eyes and helpful suggestions. Dominik Winterer wants to thank Robert Mattmüller for helpful discussions while supervising his bachelor thesis that served as a basis for this work.

References


