Stubborn Sets for Fully Observable Nondeterministic Planning

Dominik Winterer and **Robert Mattmüller**
University of Freiburg, Germany
{wintered, mattmuel}@informatik.uni-freiburg.de

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

Abstract

The stubborn set method is a state-space reduction technique, originally introduced in model checking and then transfered to classical planning. It was shown that stubborn sets significantly improve the performance of optimal deterministic planners by considering only a subset of applicable operators in a state. Fully observable nondeterministic planning (FOND) extends the formalism of classical planning by nondeterministic operators. We show that stubborn sets are also beneficial for FOND problems. We introduce nondeterministic stubborn sets, stubborn sets which preserve strong cyclic plans. We follow two approaches: Fast Incremental Planning with stubborn sets from classical planning and LAO* search with nondeterministic stubborn sets. Our experiments show that both approaches increase coverage and decrease node generations when compared to their respective baselines.

Introduction

Classical planning is the problem of finding a sequence of actions leading from a specified initial state to some goal state. Whereas in classical planning outcomes of actions are uniquely determined, fully observable nondeterministic planning (FOND) permits actions whose outcomes are uncertain. Such nondeterministic actions can be used to model, e.g., the failure of an agent's action. While this is often addressed by re-planning, strong cyclic plans—trial-and-error strategies—empower the agent to solve failure situations without re-planning.

Recently, research in classical planning has shifted towards techniques orthogonal to heuristics such as partial order reduction which has been transfered from computer aided verification (Valmari 1989; Godefroid 1995) to optimal deterministic planning (Alkhazraji et al. 2012). Further research aimed at improving the efficiency of stubborn set computation and determining a generalized definition of stubborn sets (Wehrle and Helmert 2014). We address the stubborn set method combined with two algorithms, Fast Incremental Planning (FIP) and LAO*.

Fast Incremental Planning is an algorithm for strong cyclic planning which solves FOND problems within multiple runs of an underlying classical planner (Kuter et al. 2008). Planner for Relevant Policies (PRP) combines this idea with a regression search to generalize the policy and substantially outperforms FIP (Muise, McIlraith, and Beck 2012). Our first step towards estimating the potential of stubborn sets for FOND planning is to use FIP with an underlying classical planner in combination with stubborn sets from classical planning. However, the main drawback of such determinization approaches is that they may find poor solutions, e.g., strong cyclic plans with high expected costs.

LAO* (Hansen and Zilberstein 2001), originally proposed to solve MDPs, is an algorithm for strong cyclic planning, which finds strong cyclic solutions in the nondeterministic state space. Using an admissible heuristic estimator, it finds strong cyclic plans of minimal expected costs. It has been shown that combining LAO* with pattern database heuristics (Mattmüller et al. 2010) is a successful approach to solving FOND problems. Our contribution is a stubborn set formalism for nondeterministic state spaces, that preserves strong cyclic plans. We evaluated both approaches, FIP with stubborn sets from classical planning and LAO* with our new formalism. Our results show that both approaches increase coverage and reduce node generations when compared to their respective baselines without stubborn sets.

Preliminaries

We use an SAS^+ based notation (Bäckström and Nebel 1993) to model fully observable nondeterministic planning problems. States of the world are described by a finite set of state variables $\mathcal{V}$. Every variable $v \in \mathcal{V}$ has an associated finite domain $\mathcal{D}_v$ and an extended domain $\mathcal{D}_v^+ = \mathcal{D}_v \uplus \{\perp\}$ where $\perp$ defines the undefined value. A *partial state* is a function s with $s(v) \in \mathcal{D}_v^+$ for all $v \in \mathcal{V}$. We write $vars(s)$ for the set of all v with $s(v) \neq \perp$. A partial state is a *state* if $vars(s) = \mathcal{V}$.

Definition 1 (nondeterministic planning task). *A nondeterministic planning task is a 4-tuple $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$, where $\mathcal{V}$ is a finite set of finite-domain variables, $\mathcal{O}$ is a finite set of nondeterministic operators, s_0 is a state called the initial state and s_* is a partial state called the goal. Each nondeterministic operator $o = \langle Pre \mid Eff \rangle$ has a partial state Pre called precondition, a finite set of partial states Eff and an associated non-negative number $cost(o)$ called its cost.*

An operator o is *applicable* in a state s if Pre is satisfied in s. The application of a single effect $\mathit{eff} \in \mathit{Eff}$ in s yields the state $app(\mathit{eff}, s)$ that results from updating the values of s with the ones of eff. The application of o to a state s yields the set of states $o(s) := \{app(\mathit{eff}, s) \mid \mathit{eff} \in \mathit{Eff}\}$. The set of applicable operators in a state s is denoted by $app(s)$. Sometimes we want to refer to a particular outcome of an operator $o = \langle \mathit{Pre} \mid \{\mathit{eff}_1, \cdots, \mathit{eff}_k\}\rangle$. The *determinization* of nondeterministic operator o is $o^{[1]}, \cdots, o^{[k]}$ with every outcome $o^{[i]} = \langle \mathit{Pre} \mid \{\mathit{eff}_i\}\rangle$. The *all-outcomes determinization* of planning task $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_*\rangle$ is $\Pi_{det} = \langle \mathcal{V}, \mathcal{O}_{det}, s_0, s_*\rangle$ where $\mathcal{O}_{det}$ is the set of all operator outcomes of $\mathcal{O}$.

An operator is deterministic if $|\mathit{Eff}| = 1$. It is nondeterministic if $|\mathit{Eff}| \geq 2$. We say a planning task $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_*\rangle$ is deterministic if all of its operators are deterministic. We refer to all variables in the precondition of an operator o as $prevars(o) = vars(\mathit{Pre})$ and to all variables in its effects as $\mathit{effvars} = \bigcup_i vars(\mathit{eff}_i)$.

A solution to a FOND planning task Π with set of states S is a *policy* $\pi : S \to \mathcal{O} \cup \{\bot\}$, which maps a state to an appropriate action or is undefined, e.g. $\pi(s) = \bot$. Policy π is *weak* if it defines at least one path from the initial state to a goal state following it. It is *closed* if following it either leads to a goal state or to a state where the policy is defined. Policy π is *proper* if from every state visited following it there exits a path to a goal state following it. Policy π is *acyclic* if it does not revisit already visited states.

Definition 2 (weak plan, strong cyclic plan, strong plan)**.** *Let* $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_*\rangle$ *be a planning task.*

- *A policy for* Π *is called a weak plan for* Π *if it is weak.*
- *A policy for* Π *is called a strong cyclic plan for* Π *if it is closed and proper.*
- *A policy for* Π *is called a strong plan for* Π *if it is closed proper and acyclic.*

A weak plan is a sequence of actions which leads to the goal if all nondeterministic operator outcomes were deterministic. It corresponds to a plan in classical planning. A strong plan guarantees that after a maximum number of steps a goal state is reached. Strong cyclic planning relaxes that property requiring that the goal is reached within a finite sequence of actions. We want to emphasize that the nondeterminism in FOND planning is not necessarily the same as in model checking with nondeterministic models since unlike strong cyclic plans, counterexamples in model checking are linear sequences.

Deterministic Stubborn Sets

The first step towards stubborn sets is the definition of operator interference. We follow the definition of Wehrle and Helmert (2014).

Definition 3 (interference of deterministic operators)**.** *Let* o_1 *and* o_2 *be operators of a deterministic planning task* Π *and let* s *be a state of* Π*. Operators* o_1 *and* o_2 *interfere in* s *if they are both applicable in* s*, and*

- o_1 *disables* o_2 *in* s*, i.e.,* $o_2 \notin app(o_1(s))$*, or*
- o_2 *disables* o_1 *in* s*, i.e.,* $o_1 \notin app(o_2(s))$*, or*

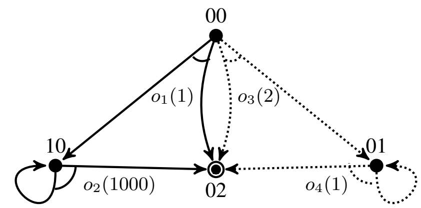

Figure 1: Solid: expensive strong cyclic solution. Dotted: cheap strong cyclic solution. Determinization-based algorithms might not find the cheap solution.

- o_1 *and* o_2 *conflict in* s*, i.e.,* $s_{12} = o_1(o_2(s))$ *and* $s_{21} = o_2(o_1(s))$ *are both defined and differ:* $s_{12} \neq s_{21}$*.*

We approximate deterministic operator interference, by considering it globally for any state s. According to this syntactic notion of interference, two deterministic operators o_1 and o_2 *interfere* if the effect of o_1 violates the precondition of o_2 (or vice versa) or if o_1 and o_2 have a common variable in their effects which they set to different values. Furthermore, we consider that operators which are never jointly applicable cannot interfere. This is done by checking whether the preconditions of two operators o_1 and o_2 are mutually exclusive (Wehrle and Helmert 2014). For stubborn sets we need two more definitions. A *disjunctive action landmark* (DAL) in state s is a set of operators such that all operator sequences leading from s to a goal state contain some operator in the set. A *necessary enabling set* (NES) for operator o in state s is a set of operators such that all operator sequences that lead from s to some goal state and include o contain some operator in the NES before the first occurrence of o. Both sets can be computed by selecting a variable v whose value differs from either the goal or the precondition of the operator to enable. Then, we add each operator which achieves the desired value of v. As both sets are not uniquely determined, the pruning power and size of stubborn sets depends on their choices (Wehrle and Helmert 2014).

Definition 4 (deterministic strong stubborn set)**.** *Let* $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_*\rangle$ *be a deterministic planning task and* s *a state. A set* $T_s \subseteq \mathcal{O}$ *is a deterministic strong stubborn set (DSSS) in* s *if the following conditions hold:*

1. T_s *contains a disjunctive action landmark in* s*.*
2. *For all operators* $o \in T_s$ *with* $o \notin app(s)$*,* T_s *contains a necessary enabling set for* o *in* s*.*
3. *For all operators* $o \in T_s$ *with* $o \in app(s)$*,* T_s *contains all operators that interfere with* o *in* s*.*

We use FIP combined with an underlying classical planner using deterministic stubborn sets. Solving FOND problems with classical planners can lead to costly strong cyclic plans. Although optimality does not play the major role in FOND planning, the possibility of finding arbitrarily bad solutions is undesirable. We show that exactly this might happen.

Example 1. Consider a nondeterministic planning task $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_*\rangle$ with variables $\mathcal{V} = \{v_1, v_2\}$ and the following operators:

- $o_1 = \langle v_1 = 0 \mid \{v_1 := 1\}, \{v_2 := 2\}\rangle$
- $o_2 = \langle v_1 = 1, v_2 = 0 \mid \{\top\}, \{v_1 := 0, v_2 := 2\}\rangle$
- $o_3 = \langle v_1 = 0 \mid \{v_2 := 2\}, \{v_2 := 1\}\rangle$
- $o_4 = \langle v_1 = 0, v_2 = 1 \mid \{\top\}, \{v_2 := 2\}\rangle$

As cost function we have $cost : \{o_1 \mapsto 1, o_2 \mapsto 1000, o_3 \mapsto 2, o_4 \mapsto 1\}$, the initial state is $s_0 = \{v_1 \mapsto 0, v_2 \mapsto 0\}$ and the goal $s_* = \{v_2 \mapsto 2\}$. Assume we perform a run of the FIP algorithm and its first weak plan would be $o_1^{[2]}$ inducing the fail-state $o_1^{[1]}(s_0) = 10$. In a subsequent weak plan search, the algorithm considers both outcomes of o_2 and adds them to the policy. This yields a clearly non-optimal strong cyclic plan, whereas the optimal solution consists of o_3 and o_4 (Figure 1). Applying PRP to this example gives the same solution, since regressing $o_1^{[2]}$ is ineffective.

Nondeterministic Stubborn Sets

Reducing FOND problems to multiple classical planning problems sometimes leads to poor strong cyclic solutions since the individual runs of classical planners only guarantee good weak plans which are not always part of a good strong cyclic plans. To overcome this, it can be beneficial to plan in the nondeterministic state space e.g., with LAO* search (Hansen and Zilberstein 2001) which finds strong cyclic plans with minimum expected costs under certain assumptions. Planning in the nondeterministic state space needs new definitions of stubborn sets and operator interference since the former do not consider nondeterministic operators.

For a given nondeterministic planning problem Π, a straightforward approach would be to directly apply the original definition of strong stubborn sets on the all-outcome-determinization of Π, and additionally, to add for every outcome $o^{[i]}$ of a nondeterministic operator o every other outcome of o in order to respect o's nondeterministic nature. However, as the following example shows, such an approach is incomplete.

Example 2. Consider the following all-outcomes determinization $\Pi_{det} = \langle \mathcal{V}, \mathcal{O}_{det}, s_0, s_* \rangle$ of nondeterministic planning task Π with variables $\mathcal{V} = \{v_1, v_2\}$ and the following operators:

- $o_1^{[1]} = \langle v_1 = 0 \mid \{v_1 := 1\}\rangle$, $o_1^{[2]} = \langle v_1 = 0 \mid \{v_1 := 2\}\rangle$
- $o_2^{[1]} = \langle v_2 = 0 \mid \{v_2 := 1\}\rangle$, $o_2^{[2]} = \langle v_2 = 0 \mid \{v_2 := 2\}\rangle$
- $o_3^{[1]} = \langle v_2 = 0 \mid \{v_2 := 3\}\rangle$, $o_3^{[2]} = \langle v_2 = 0 \mid \{v_2 := 4\}\rangle$
- $o_{11} = \langle v_1 = 1, v_2 = 1 \mid \{v_2 := 5\}\rangle$
- $o_{12} = \langle v_1 = 1, v_2 = 2 \mid \{v_2 := 5\}\rangle$
- $o_{23} = \langle v_1 = 2, v_2 = 3 \mid \{v_2 := 5\}\rangle$
- $o_{24} = \langle v_1 = 2, v_2 = 4 \mid \{v_2 := 5\}\rangle$

The initial state is $s_0 = \{v_1 \mapsto 0, v_2 \mapsto 0\}$, and the goal is $s_* = \{v_2 \mapsto 5\}$. The set $\{o_{11}, o_{12}, o_{23}, o_{24}\}$ is a disjunctive action landmark in s_0 which we add to the candidate 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

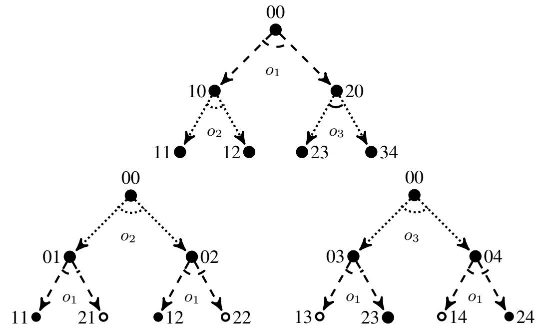

Figure 2: Postponing o_1 in s_0 only leads to policies with dead-ends. The straightforward instantiation is incomplete.

$v_2 = 4$ in the preconditions of o_{11}, o_{12}, o_{23}, o_{24}, respectively, and to add the determinized 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 non-interfering with any operator not in T_{s_0}. Hence, we finally get $T_{s_0} = \{o_{11}, o_{12}, o_{23}, o_{24}, o_2^{[1]}, o_2^{[2]}, o_3^{[1]}, o_3^{[2]}\}$.

However, T_{s_0} is insufficient for our purpose because every strong plan from s_0 has to start with o_1: Depending on the nondeterministic outcome of o_1 ($v_1 = 1$ 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 2).

The core problem of our straightforward instantiation is that deterministic operator interference is an insufficient criterion for nondeterministic operators. Because changing the order of two non-interfering nondeterministic operators o and o' in a strong cyclic plan results in, e.g., outcomes of o' getting prefixes of weak plans started by o. While this is not an issue for all weak plans which contain outcomes of both operators, it is problematic to those weak plans which start with an outcome of o but do not contain an outcome of o'. A solution to this is to demand that such prefixes preserve the original weak plan which we address with the following property.

Definition 5 (prefix-compatibility)**.** *Let Π be a planning task and $\Pi_{det} = \langle \mathcal{V}, \mathcal{O}_{det}, s_0, s_* \rangle$ its all-outcomes determinization. Two operators $o_1, o_2 \in \mathcal{O}_{det}$ are* prefix compatible *if for all operator sequences π_1 and π_2:*

- *$o_1\pi_1$ is a weak plan implies $o_2o_1\pi_1$ is also a weak plan and*
- *$o_2\pi_2$ is a weak plan implies $o_1o_2\pi_2$ is also a weak plan*

Intuitively, two operators o_1 and o_2 are prefix compatible if every weak plan starting with o_1 is preserved if we put o_2 to its front and vice versa. Equipped with prefix compatibility, we can formulate the definition of a stubborn set for the nondeterministic state space which has two additional rules compared to the DSSS definition.

Definition 6 (nondeterministic strong stubborn set)**.** *Let $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a nondeterministic planning task,*

$\Pi_{det} = \langle \mathcal{V}, \mathcal{O}_{det}, s_0, s_* \rangle$ *its all-outcomes determinization and s a state. A set $T_s \subseteq \mathcal{O}_{det}$, is a nondeterministic strong stubborn set (NSSS) in s if the following conditions hold:*

1. *T_s contains a disjunctive action landmark in s for Π_{det}.*
2. *For all operators $o \in T_s$ with $o \notin app(s)$, T_s contains a necessary enabling set for o in s for Π_{det}.*
3. *For all operators $o \in T_s$ with $o \in app(s)$, T_s contains all operators that interfere with o in s for Π_{det}.*
4. *For every outcome $o^{[i]} \in T_s$ of nondeterministic operator o, T_s contains all operators that are not prefix compatible with o.*
5. *For every outcome $o^{[i]} \in T_s$ of nondeterministic operator o, T_s contains all other outcomes of o.*

Proposition 1. *Nondeterministic strong stubborn sets preserve completeness for strong cyclic planning.*

Proof. At first we show completeness for strong planning then we show it for strong cyclic planning. Let π be a strong plan from state s that induces weak plans π_i and π_j, such that there is state $\tilde{s}$ with $\pi_i(\tilde{s}) = o^{[i]}$ and $\pi_j(\tilde{s}) = o^{[j]}$. Weak plans π_i and π_j have the following structure: $\pi_i = \alpha o^{[i]} \beta_i$ and $\pi_j = \alpha o^{[j]} \beta_j$ where α is a common operator sequence without outcomes of nondeterministic operators, and β_i, β_j contain also outcomes of nondeterministic operators respectively. State $\tilde{s}$ is the branching point of π_i and π_j. Let k_i be the smallest index such that operator $o_{k_i} \in \pi_i$ is contained in the nondeterministic stubborn set T_s, similarly for k_j and π_j. We distinguish the following three cases.

(1) $o_{k_i} \in \alpha = o_1 \cdots o_n$. Clearly $o_{k_i} = o_{k_j}$. o_{k_i} is applicable since otherwise a necessary enabling set has to be contained in T_s and at least one operator has to be applied before o_{k_i}, contradicting the choice of k_i. Since k_i is the smallest index such that $o_{k_i} \in T_s$, o_{k_i} does not interfere with any operator of smaller index because otherwise an operator applied before o_{k_i} must be contained in T_s. Also, this contradicts the choice of k_i. Thus we can replace α by $o_{k_i} o_1 \cdots o_{k_i - 1} o_{k_i + 1} \cdots o_n$.

(2) $o_{k_i} = o^{[i]}$. $o_{k_j} \notin \alpha$ since otherwise $o_{k_i} \in \alpha$. Also, o_{k_j} cannot be in β_j because by the definition of the NSSS $o^{[j]} \in T_s$. It follows that $o_{k_j} = o^{[j]}$. Like in case (1) o_{k_i} is applicable and does not interfere with operators of smaller index for π_i, the same holds for o_{k_j} and π_j. Thus we can move the nondeterministic operator o to the front resulting in $o^{[i]} \alpha \beta_i$ and $o^{[j]} \alpha \beta_j$.

(3) $o_{k_i} \in \beta_i = o_{n+2} \cdots o_{n+m_i}$. o_{k_j} cannot be in α since otherwise $o_{k_i} \in \alpha$. Also, $o_{k_j} \neq o^{[j]}$ since otherwise by definition of the NSSS, $o^{[i]}$ would be included in T_s, contradicting the choice of o_{k_i}. Therefore $o_{k_j} \in \beta_j$. Let $s_1 \cdots s_{n-1} \tilde{s} s_{n+1} \cdots s_{n+m_i}$ be the states visited by π_i. We know as in case (1) that o_{k_i} does not interfere with operators of smaller index. Inductively it follows that o_{k_i} is applicable in $\tilde{s}$. Also we know that o_{k_i} and o are prefix compatible since otherwise $o \in T_s$. This means that $o^{[i]} o_{n+2} \cdots o_{n+m_i}$ is a weak plan from $o_k(\tilde{s}) = o_k(o_n \cdots (o_1(s_0)))$. From the non-interference of o_k with operators of smaller index we get $o_k(o_n \cdots (o_1(s_0))) = o_n(o_{n-1} \cdots (o_1(o_k(s_0))))$. Hence we can move o_k to the front of π_i and π_j. If o_k is an outcome of a nondeterministic operator o' then it has a sibling o'^{l} which is the smallest index of weak plan π_l. This case is covered by case (2) with π_l and π_i.

Since a strong plan is a strong cyclic plan without cycles we just have to consider the effect of cycles. NSSS is state dependent but not path dependent, therefore revisiting some state s does not affect T_s, concluding the proof. □

Nondeterministic stubborn sets are in general not optimality preserving for strong cyclic planning since prefix compatibility leads to operators being added in front of other ones which can lead to solutions with higher expected costs.

Approximating Prefix Compatibility

The exact notion of prefix compatibility is intractable to compute because we would have to consider all weak plans. Therefore we outline how to find a sufficient criterion for prefix-compatibility. We define $Dis(o)$ as the set of operator-variable pairs $(o', v) \in \mathcal{O}_{det} \times \mathcal{V}$ such that o disables o' on variable v in any state. Further we define $Neg(o)$ as the set of goal variables with which o conflicts, i.e., $\mathit{eff}(o)[v] \neq s_*[v]$ for goal-related variables v on which o has an effect. If $Dis(o_1) = Dis(o_2)$ and $Neg(o_1) = Neg(o_2)$ then o_1 and o_2 are prefix compatible. The idea behind this is: if two operators o_1 and o_2 disable the same set of operators on the same variables, then every deterministic operator sequence starting with o_1 remains applicable if we append o_2 to its front. Also weak plans are preserved since o_1 and o_2 do not violate different goal variables.

In some cases, we can weaken this syntactic notion of prefix compatibility. Consider two non-interfering operators $o_1 = \langle Pre \mid \{\mathit{eff}'_1\} \rangle$ and $o_2 = \langle Pre \mid \{\mathit{eff}_1, \cdots, \mathit{eff}_n\} \rangle$. If $\sigma = \{s \mapsto o_1, o_1(s) \mapsto o_2\}$ is a subsequence of a strong cyclic plan from state s then exchanging the order of o_1 and o_2 gives an equivalent subsequence since they induce the same set of states, i.e., $o_1(o_2^{[i]}(s))) = o_2^{[i]}(o_1(s))$ for all $i \leq n$. Therefore if two such operators do not interfere, it suffices to check $Dis(o_1) \subseteq Dis(o_2)$ and $Neg(o_1) \subseteq Neg(o_2)$.

Sometimes nondeterministic operators contain only one nontrivial effect, i.e. an operator $o = \langle Pre \mid \{\mathit{eff}_1\}, \{\top\} \rangle$. For every weak plan $o^{[1]}\pi$ from state s, it exists a finite sequence $\sigma = o^{[2]} \cdots o^{[2]}$, repeated applications of o's trivial effect, such that $\sigma o^{[1]} \pi$ is also a weak plan from s. Thus, every operator being a prefix of $o^{[1]}\pi$ preserving the weak plan, does also preserve $\sigma o^{[1]} \pi$. Such operators are therefore trivially prefix compatible to any other deterministic operator.

Efficient Computation

As nondeterministic stubborn sets leave open how the disjunctive action landmark and the necessary enabling sets were chosen, the pruning power of stubborn sets depends highly on these design choices. We outlined how prefix compatibility can be syntactically addressed. However, for applicable nondeterministic operators with more than one nontrivial effect, in the stubborn set we have to add both the interfering and the non-prefix compatible operators. This leads to many operators being added to the stubborn set. It is therefore reasonable to avoid applicable nondeterministic operators with more than one nontrivial effect from being added to the stubborn set. Let $nontrivial$ be the set of operators with more than one nontrivial effect. Our intention is to exclude applicable operators of $nontrivial$ from being added to the stubborn set. We address this by computing a weight whenever we have to add a DAL or NES to the stubborn set. We calculate a weight for each DAL or NES and chose the DAL or NES with lowest weight according to:

$$weight(o, s, T_s) = \begin{cases} \infty, & \text{if } o \in app(s) \wedge o \in nontrivial \\ K, & \text{if } o \in app(s) \wedge o \notin nontrivial \\ 1, & \text{otherwise} \end{cases}$$

where o is an operator not in stubborn set T_s and K a nonzero natural number. Our *exclude* strategy is an extension of a strategy presented by Laarman et al. (2013) which penalizes applicable operators not in the current candidate stubborn set. A coarser strategy towards prefix compatibility for nontrivial operators is to simply assume that an applicable nontrivial operator is not prefix compatible to all other operators. On par with the exclude strategy this is feasible since it avoids the costly computation of the disabling relation.

A Tighter Envelope

Active operators (Chen and Yao 2009; Wehrle et al. 2013) approximate the set of operators which can be part of any weak plan from some state using domain transitions graphs (DTGs). From a more general point of view, Wehrle et al. (2013) denote subsets which preserve at least one weak plan from some state as an *envelope*. Combining a tight envelope with stubborn sets may not only exclude operators which are not in envelope E from the stubborn set but also prevent cascades from being added to the stubborn set. Of course, the active operators can also be used for strong cyclic planning since strong cyclic plans consist of multiple weak plans. We additionally exploit the structure of strong cyclic plans and obtain a tighter envelope.

A *part-of-a-plan operator* $o \in \mathcal{O}$ in s is a deterministic operator that is contained in some weak plan starting from s. This notion is intractable to compute so we have to find a sufficient criterion for it.

Definition 7 (active operator). *Let $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_* \rangle$ be a deterministic planning task. An active operator $o \in \mathcal{O}$ in a state s is an operator that satisfies the following conditions:*

1. *For every variable $v \in prevars(o)$, there is a path in $DTG(v)$ from $s[v]$ to $pre(o)[v]$, and also from $pre(o)[v]$ to the goal value $s[v]$ if v is goal-related.*
2. *For all $v \in effvars(o) \cap vars(s_*)$ there is a path in $DTG(v)$ from $eff_1(o)[v]$ to $s_*[v]$.*

Intuitively, the definition states that an operator is active if it is part of some weak plan from the corresponding abstracted state in every singleton abstraction of Π. Thus, a part-of-a-plan operator is always active but not vice versa.

A *nondeterministic part-of-a-plan operator* $o \in \mathcal{O}$ in s is an operator that is contained in some strong plan π starting from s. Like for the part-of-a-plan operators this is intractable to compute.

Proposition 2. *Let o be an applicable nondeterministic operator and $o^{[i]}$ one of its outcomes. If $o^{[i]}$ is inactive in state s then o cannot be part of any strong cyclic plan from s.*

Proof. We show this by contradiction. If there were a strong cyclic plan π from s, such that $\pi(\tilde{s}) = o$ for some state $\tilde{s}$ but $o^{[i]}$ is no part-of-a-plan operator in s. Let further $\sigma = o_1 \cdots o_n$ be a deterministic operator sequence applicable in s that leads to $\tilde{s}$. Since $o^{[i]}$ is no part-of-a-plan operator in s, no weak plan from s does contain $o^{[i]}$. Therefore $\sigma o^{[i]} \pi'$ for an arbitrary operator sequence π' cannot be a weak plan from s. This implies that π' is no weak plan from $o^{[i]}(\tilde{s})$. Thus π is not proper since $o(\tilde{s})$ is reachable following π but there is no goal state reachable from $o(\tilde{s})$. This contradicts π being a strong plan. It follows that if any outcome of a nondeterministic operator o is not active in s, then o cannot be part of a strong cyclic plan from s. □

We denote our new envelope by *nondeterministic active envelope*. It can be used for both the DSSS and NSSS.

Experimental Evaluation

We focused our experimental evaluation on the following two configurations:

1. FIP combined with DSSSs
2. LAO* combined with NSSSs

We further investigated the impact of different envelopes: *full, active, nondeterministic active* (Table 1 and Table 2). Also, we varied the approximation of prefix compatibility for the NSSS approach (Table 3). We differentiate between the approach which assumes that every nontrivial operator is not prefix compatible with all other operators (*no prefix*) and the approach where prefix compatibility is syntactically approximated (*syntactic*). For the DSSS, disjunctive action landmarks and necessary enabling sets were computed using the *laarman* strategy. For the NSSS we used the *exclude* strategy. The interference relation for both the DSSSs and NSSSs is entirely precomputed which is also true for the achievers, the NSSSs need the additional precomputation of the disabling relation. For the underlying classical planner of FIP, we used greedy best first search. As heuristic estimator, we chose the FF heuristic (Hoffmann and Nebel 2001) for all approaches.

We evaluated both stubborn set approaches on all FOND domains of the IPC-2008 and variations of these. Furthermore we added two domains from probabilistic planning to our benchmark set.[1] All experiments were conducted on a server equipped with AMD Opteron 2.3 GHz CPUs. We set

[1] *First-Responders-new* consists of larger instances of the First-Responders domain. *Forest-new* is taken by Muise, McIlraith, and Beck (2012). *Tidyup* is the *Mobile Manipulation* domain of Hertle et al. (2014) adapted for FOND planning. *Earth-Observation* was introduced by Aldinger and Löhr (2013).

	Coverage				Node Generations			
Domain	FIP	DSSS	DACT	NACT	FIP	DSSS	DACT	NACT
FR(75)	**74**	–1	–1	–1	2917345	45.71%	45.71%	**45.65%**
FR-NEW(91)	**75**	–1	–1	–1	9849763	65.3%	65.3%	**65.29%**
FOREST-NEW(90)	16	**+4**	**+4**	**+4**	16463519	**10.94%**	**10.94%**	**10.94%**
FOREST(90)	13	**+4**	**+4**	**+4**	28857372	**18.96%**	**18.96%**	**18.96%**
EARTH(40)	**35**	–2	–2	–2	**5592159**	**100.0%**	**100.0%**	**100.0%**
TIDYUP(10)	5	±0	**+5**	**+5**	2307584	99.94%	0.12%	**0.1%**
TTIREWORLD(40)	**3**	**±0**	**±0**	**±0**	**24805**	**100.0%**	**100.0%**	**100.0%**
BW(30)	**25**	–1	–1	–1	**191627**	**100.0%**	**100.0%**	**100.0%**
FAULTS(55)	**55**	**±0**	**±0**	**±0**	**160599**	**100.0%**	**100.0%**	**100.0%**
Overall	301	+3	**+8**	**+8**	66364773	35.13%	31.66%	**31.65%**

Table 1: Comparison of plain FIP with FIP using DSSS and different envelopes: full, active (DACT), nondeterministic active (NACT). Nodes of DACT, NACT in % of plain FIP.

	Coverage				Node Generations			
Domain	LAO*	NSSS	DACT	NACT	LAO*	NSSS	DACT	NACT
FR(75)	57	+1	+1	**+2**	323275	85.86%	85.86%	**81.51%**
FR-NEW(91)	**19**	–2	–2	–2	**455521**	100.7%	100.7%	100.77%
FOREST-NEW(90)	3	**+3**	+2	**+3**	10989	**93.28%**	**93.28%**	**93.28%**
FOREST(90)	6	**+1**	**+1**	**+1**	**17861**	100.96%	100.96%	100.96%
EARTH(40)	**30**	**±0**	**±0**	**±0**	**80229**	**100.0%**	**100.0%**	**100.0%**
TIDYUP(10)	**9**	**±0**	**±0**	**±0**	40166	99.95%	72.47%	**59.68%**
TTIREWORLD(40)	**6**	**±0**	**±0**	**±0**	**27807**	**100.0%**	**100.0%**	**100.0%**
BW(30)	**21**	**±0**	**±0**	**±0**	**111845**	**100.0%**	**100.0%**	**100.0%**
FAULTS(55)	**54**	**±0**	**±0**	**±0**	**153582**	**100.0%**	**100.0%**	**100.0%**
Overall	205	+3	+2	**+4**	1221275	96.47%	95.57%	**94.02%**

Table 2: Comparison of plain LAO* with LAO* using NSSS and different envelopes: full, active (DACT), nondeterministic active (NACT). Nodes of DACT, NACT in % of plain LAO*.

a time limit of 30 minutes and a memory limit of 4GB for the Java Runtime Environment. We considered an instance as solved if the planner finds a solution within its time limit or proves that none exists.

Results

Our experiments yield the following insights:

- *DSSS approach*: FIP combined with DSSS significantly increases coverage and reduces node expansions. This is particularly pronounced in the domains *Forest* and *Forest-new*. DSSS with active envelope solves five additional problem instances of the *Tidyup* domain which has—in contrast to all other domains—applicable inactive operators permitting immense pruning (0.1% nodes generated). In all other domains the effect of active envelope and also of nondeterministic active envelope is negligible. In *First-Responders* and *First-Responders-new* the DSSS approach loses one instance because of heuristic tie-breaking. Also, it loses two instances in *Earth-Observation* and one in *Blocksworld*. The instances in *Earth-Observation* result from the vast number of generated states. In *Blocksworld* the DSSS approach fails to solve the hardest problem solved by the baseline which solves it close to the time limit (1564s out of 1800s).
- *NSSS approach*: LAO* search combined with NSSS does also clearly outperform its baseline in terms of coverage and node expansions. But in contrast to the DSSS approach, node generations are not as drastically reduced. Since the LAO* algorithm produces noticeably more overhead per node than the FIP algorithm, reducing a single node has greater impact for the LAO* algorithm than for FIP. The coverage increase is most evident in the domain *Forest-new* where three additional instances are solved and *First-Responders-new* with two additionally solved instances. The loss of two instances in *First-Responders-new* is caused by heuristic tie-breaks (100.7% node expansions). A blind search without stubborn set does not solve any problem in this domain, whereas in combination with NSSS, it solves the smallest instance. Furthermore, active envelope and nondeterministic active envelope are beneficial in terms of pruning power. We again see the empirical dominance of nondeterministic active envelope over active envelope supporting the theoretical results.
- *Prefix Compatibility*: In terms of pruning power, the approach without prefix compatibility for the nontrivial operators is dominated by the syntactic approximation of prefix compatibility. Except in the domain *First-Responders-new* more nodes are generated because of a single instance which is caused by heuristic tie-breaking (100.45% node expansions). The increased pruning power is reflected in better coverage for *First-Responders-new* and *Forest-new* where two hard additional instances are solved respectively.

	Coverage		Node Generations	
Domain	no prefix	syntactic	no prefix	syntactic
FR(75)	**59**	–2	264437	**96.27%**
FR-NEW(91)	17	**+1**	**535598**	100.45%
FOREST-NEW(90)	6	**+1**	21088	**56.02%**
FOREST(90)	**7**	–1	18033	**72.36%**

Table 3: Comparison of no prefix compatibility for nontrivial operators with syntactic approximation. We grouped the domains where coverage and node generations are equal. Nodes of syntactic in % of no prefix approach.

Conclusion

We demonstrated that the stubborn set approach is also beneficial for FOND planning. While this was expectable for FIP and deterministic strong stubborn sets, we needed a new formalism for LAO* search, which does not reduce FOND to classical planning. We provided a novel notion of stubborn sets and proved that it is completeness preserving.

For the future, we want to focus on how prefix compatibility can be better approximated and evaluate how the degree of a nondeterministic operator affects the results of our approach. As a first step, we noticed that two operators cannot be prefix compatible if they have mutex preconditions. Also, it would be interesting to combine PRP with stubborn sets.

Acknowledgments

This work was partly supported by the DFG as part of the SFB/TR 14 AVACS and by the Swiss National Science Foundation (SNSF) as part of the project "Automated Reformulation and Pruning in Factored State Spaces (ARAP)".

References

Aldinger, J., and Löhr, J. 2013. Planning for agile earth observation satellites. In *Proceedings of the Workshop on Planning in Continuous Domains (PCD) at ICAPS 2013*.

Alkhazraji, Y.; Wehrle, M.; Mattmüller, R.; and Helmert, M. 2012. A stubborn set algorithm for optimal planning. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, 891–892.

Bäckström, C., and Nebel, B. 1993. Complexity results for sas+ planning. *Computational Intelligence* 11:625–655.

Chen, Y., and Yao, G. 2009. Completeness and optimality preserving reduction for planning. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI 2009)*, 1659–1664.

Godefroid, P. 1995. Partial-order methods for the verification of concurrent systems: An approach to the state-explosion problem.

Hansen, E. A., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129(1–2).

Hertle, A.; Dornhege, C.; Keller, T.; Mattmüller, R.; Ortlieb, M.; and Nebel, B. 2014. An experimental comparison of classical, FOND and probabilistic planning. In *Proceedings of the 37th Annual German Conference on AI (KI 2014)*, 297–308.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (JAIR)* 14:2001.

Kuter, U.; Nau, D.; Reisner, E.; and Goldman, R. P. 2008. Using classical planners to solve nondeterministic planning problems. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS 2008)*, 190–197.

Laarman, A.; Pater, E.; van de Pol, J.; and Weber, M. 2013. Guard-based partial-order reduction. In *Proceedings of the 21st International Symposium on Model Checking of Software (SPIN 2014)*, 227–245.

Mattmüller, R.; Ortlieb, M.; Helmert, M.; and Bercher, P. 2010. Pattern database heuristics for fully observable nondeterministic planning. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS 2010)*, 105–112.

Muise, C.; McIlraith, S. A.; and Beck, J. C. 2012. Improved non-deterministic planning by exploiting state relevance. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*, 172–180.

Valmari, A. 1989. Stubborn sets for reduced state space generation. In *Proceedings of the Tenth International Conference on Application and Theory of Petri Nets (APN 1989)*, 1–22.

Wehrle, M., and Helmert. 2014. Efficient stubborn sets: Generalized algorithms and selection strategies. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 323–331.

Wehrle, M.; Helmert, M.; Alkhazraji, Y.; and Mattmüller, R. 2013. The relative pruning power of strong stubborn sets and expansion core. In *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS 2013)*, 251–259.