ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-1097

Bounded Intention Planning Revisited

Silvan Sievers and **Martin Wehrle** and **Malte Helmert**[1]

1 INTRODUCTION

Bounded Intention Planning (BIP) [7] is a pruning approach for optimal planning with unary operators. BIP has the flavor of partial order reduction, which has recently found increasing interest for optimal planning [4, 6]. However, although BIP is claimed to be a variant of *stubborn sets* [3] in the original paper, no proof is given for this claim. In this paper, we shed light on the relationship of BIP and stubborn sets. In particular, we show that BIP's operator partitions *sometimes* correspond to strong stubborn sets defined for planning [5].

2 PRELIMINARIES

We consider SAS^+ planning with action costs. A *planning task* is a 4-tuple $\Pi = \langle \mathcal{V}, \mathcal{O}, s_0, s_\star \rangle$, where $\mathcal{V}$ is a finite set of *state variables*, $\mathcal{O}$ is a finite set of *operators*, s_0 is the *initial state* and $s_\star$ is the *goal*. Every $v \in \mathcal{V}$ has a finite domain $\mathcal{D}(v)$. A variable assignment on a subset of $\mathcal{V}$ is called a *partial state* s; we denote the set of variables mentioned in s by $vars(s)$. A partial state is a *state* if $vars(s) = \mathcal{V}$. By $s[v]$ we refer to the value of v in s. A (partial) state s *complies* with a (partial) state s' iff $s[v] = s'[v]$ for all $v \in vars(s) \cap vars(s')$. The initial state s_0 is a state and $s_\star$ is a partial state. Every $o \in \mathcal{O}$ has a *precondition* pre_o, an *effect* eff_o and a *prevail-condition* prv_o, which are partial states, and associated *cost* $c(o) \in \mathbb{R}_0^+$. If $v \in vars(eff_o)$, then $v \notin vars(prv_o)$ and possibly $v \in vars(pre_o)$; otherwise $v \notin vars(pre_o)$ and possibly $v \in vars(prv_o)$. An operator o is *applicable* in s if both pre_o and prv_o comply with s. The result of applying o in s is the *successor state* s' that complies with eff_o and satisfies $s'[v] = s[v]$ for all $v \notin vars(eff_o)$. A sequence of operators $o_1, \ldots, o_n \in \mathcal{O}$ is called an *s-plan* if applying all operators in sequence, starting at state s, results in a state complying with $s_\star$. A *plan* for a task Π is defined as an s_0-plan for Π. A plan is *optimal* if its cost $\sum_{i=1}^{n} c(o_i)$ is minimal among all plans. The objective of optimal planning is to find an optimal plan or to prove that no plan exists.

2.1 BOUNDED INTENTION PLANNING

We introduce the essential parts of bounded intention planning (BIP), which we will relate to stubborn sets afterwards.

BIP is defined for *unary* planning tasks ($|vars(eff_o)| = 1$ for all $o \in \mathcal{O}$). Roughly speaking, BIP augments the original planning task Π with *intention variables* and defines several "intermediate" operators for each original operator. The resulting augmented planning task $\bar{\Pi} = (\bar{\mathcal{V}}, \bar{\mathcal{O}}, \bar{s_0}, \bar{s_\star})$ can then be exploited for pruning. Let *CG* be the *causal graph* [1] of Π, which is a directed graph with nodes $\mathcal{V}$ and edges from v to w iff there exists an operator $o \in \mathcal{O}$ with $v \in prv_o$ and $w \in eff_o$ (recall that $\mathcal{O}$ only contains unary operators, so there are no edges between effect variables). We denote the successors of v in *CG* by $CG(v)$. Furthermore, let $\mathcal{O}[v] \subseteq \mathcal{O}$ denote the operators o with $v \in vars(eff_o)$.

For every $v \in \mathcal{V}$, the augmented variable set $\bar{\mathcal{V}}$ contains v and two additional intention variables O_v and C_v. O_v has domain $\mathcal{D}(O_v) = \mathcal{O}[v] \cup \{free, frozen\}$, and C_v domain $\mathcal{D}(C_v) = CG(v) \cup \{free\}$.

For all $v \in \mathcal{V}$, the augmented operator set $\bar{\mathcal{O}}$ contains the following operators: first, for every $o \in \mathcal{O}[v]$, there is a "set operator intention" operator $SetO(o)$ with $pre[O_v] = free$, $eff[O_v] = o$ and $prv[v] = pre_o[v]$, with cost $c(o)$; second, for every $x \in \mathcal{D}(v)$, there is a zero-cost "freeze operator intention" operator $Freeze(v, x)$ with $pre[O_v] = free$ and $eff[O_v] = frozen$; third, for every $v \in \mathcal{V}$ and $c \in CG(v)$, there is a zero-cost "set child intention" operator $SetC(v, c)$ with $pre[C_v] = free$ and $eff[C_v] = c$; fourth, for every $o \in \mathcal{O}[v]$, there is a zero-cost "fire" operator $Fire(o)$ which has the same conditions and effects as o and in addition $pre[O_v] = o$, $eff[O_v] = free$, and for all $w \in vars(prv_o)$, $pre[O_w] = frozen$, $eff[O_w] = free$ and $pre[C_w] = v$, $eff[C_w] = free$.

The augmented initial state $\bar{s_0}$ extends s_0 by setting all new O_v and C_v variables to *free*. The augmented goal $\bar{s_\star}$ is the same as $s_\star$.

BIP *partitions* the operators into partitions of three types: for each $v \in \mathcal{V}$ and $x \in \mathcal{D}(v)$, there is a partition $\mathrm{SetO}_{v=x} = \{SetO(o) \mid o \in \mathcal{O}[v] \wedge pre_o[v] = x\} \cup \{Freeze(v, x)\}$; for each $v \in \mathcal{V}$, there is a partition $\mathrm{SetC}_v = \{SetC(v, c) \mid c \in CG(v)\}$; for each $o \in \mathcal{O}$, there is a partition $\mathrm{Fire}_o = \{Fire(o)\}$.

Let P denote the set of all such partitions. By definition, either *all* operators in a partition are applicable or *none*. We denote the set of applicable partitions for a given state s with P_s. The central theorem of Wolfe and Russell [7] states that we can choose a *single, arbitrary* partition from P_s in every state s and still preserve optimality. Branching is restricted to operators within this partition.

2.2 STUBBORN SETS

Stubborn sets were introduced by Valmari [3]. To state the definition (adapted to planning tasks), we need the concept of *necessary enabling sets (NES)*. For a state s and operator o not applicable in s, a NES for o and 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 from the NES before the first occurrence of o.

Operator o *disables* operator o' if there is a variable v with values $\{x, x'\} \subseteq \mathcal{D}(v)$ such that $x \neq x'$, $eff_o[v] = x$ and either $prv_{o'}[v] = x'$ or $pre_{o'}[v] = x'$. Operators o and o' *interfere* if o disables o', or vice versa, or $eff_o[v] = x$ and $eff_{o'}[v] = x'$ for $x' \neq x$.

Definition 1. *A set of operators $T_s \subseteq \mathcal{O}$ of a task Π is a* strong semistubborn set *in state s iff for all $o \in T_s$ not applicable in s, T_s contains a necessary enabling set for s, and for all $o \in T_s$ applicable in s, T_s contains all operators that interfere with o.*

[1] University of Basel, Switzerland, firstname.lastname@unibas.ch

A set of operators $T_s \subseteq \mathcal{O}$ is a strong stubborn set in the Valmari sense *iff T_s is a strong semistubborn set, and T_s contains at least one applicable operator in s if such an operator exists.*

Strong stubborn sets in the Valmari sense guarantee to preserve deadlocks, but not goal reachability. However, goal reachability can be reduced to deadlock detection, which yields corresponding definitions for planning tasks [4, 5]. We provide the definition of generalized strong stubborn sets given by Wehrle and Helmert [5], simplified to the setting needed for this paper. A state s is called *solvable* iff there exists an s-plan. Furthermore, following Wehrle and Helmert, we call a plan π *strongly optimal* iff π is an optimal plan with minimum number of zero-cost operators.

Definition 2. *A set of operators $T_s \subseteq \mathcal{O}$ of a task Π is a* generalized strong stubborn set (GSSS) in the planning sense *in the solvable state s iff T_s is a strong semistubborn set in s, and T_s contains at least* one *operator from at least* one *strongly optimal plan starting in s.*

3 RELATION TO STUBBORN SETS

BIP's applicable operator partitions induce strong semistubborn sets that contain exactly the same applicable operators.

Theorem 1. *Let s be a state, $X \in P_s$ be an applicable partition. Then $T_s := X \cup \{o \mid o \text{ interferes with } o' \in X\}$ is a strong semistubborn set with the same applicable operators as X.*

Proof sketch. We exemplarily prove the claim for the case that X is a partition $\text{Fire}_o = \{\mathit{Fire}(o)\}$. All remaining cases follow a similar argumentation. The full proof can be found in a technical report [2].

Without loss of generality, assume $v \in \mathit{vars}(\mathit{eff}_o)$. We have to show that all operators that interfere with $\mathit{Fire}(o)$ are not applicable in s and that T_s already contains a NES for those operators. We show the claim for the most involved case where an operator $\mathit{Fire}(o')$ interferes with $\mathit{Fire}(o)$. Let us assume that $\mathit{Fire}(o')$ disables $\mathit{Fire}(o)$ via variable $w \in \mathcal{V}$, $w \neq v$, with $\mathit{eff}_{o'}[w] \neq \mathit{prv}_o[w]$.

We claim that $\mathit{Fire}(o')$ is not applicable in s. By the definition of *Fire* operators, we have $\mathit{pre}_{\mathit{Fire}(o)}[O_w] = \mathit{frozen}$ and thus $s[O_w] = \mathit{frozen}$ because $\mathit{Fire}(o)$ is applicable in s. Again by definition, we have $\mathit{pre}_{\mathit{Fire}(o')}[O_w] = o' \neq s[O_w]$, proving the claim.

Furthermore, we claim that $\{\mathit{Fire}(o)\}$ is a NES for $\mathit{Fire}(o')$ in s. We observe that the value of O_w must change from *frozen* to *free* before it can be set to o' as required by $\mathit{Fire}(o')$. Only some operator $\mathit{Fire}(\hat{o})$ with $w \in \mathit{vars}(\mathit{prv}_{\hat{o}})$ can set O_w to *free*. Let $v' \in \mathit{vars}(\mathit{eff}_{\hat{o}})$. If $v' = v$, $\mathit{pre}_{\mathit{Fire}(\hat{o})}[O_v] = \hat{o} \neq o = s[O_v]$ and thus $\mathit{Fire}(o)$ must be applied first. If $v' \neq v$, $\mathit{pre}_{\mathit{Fire}(\hat{o})}[C_w] = v' \neq v = s[C_w]$ and only some operator $\mathit{Fire}(\hat{o}')$ with $v \in \mathit{vars}(\mathit{eff}_{\hat{o}'})$ and $w \in \mathit{vars}(\mathit{prv}_{\hat{o}'})$ can change C_w from v to *free* (required before setting it to v'). Because $s[O_v] = o$, this must be $\mathit{Fire}(o)$, proving the claim. □

As the induced semistubborn set T_s contains an applicable operator partition by definition, it contains at least one applicable operator.

Corollary 1. *Let s be a state, $X \in P_s$ be an applicable operator partition. Then T_s induced by X defined in Theorem 1 is a strong stubborn set in the Valmari sense.*

As shown by Wolfe and Russell, every applicable operator partition starts an optimal plan. However, not all such partitions contain an operator that starts a *strongly* optimal plan[2], which is the missing criterion for T_s to be a GSSS in the planning sense (cf. Definition 2). Nevertheless, there always exists at least *one* partition in P_s which induces a GSSS in the planning sense.

[2] Consider a task with variables v and w, initially 0, goal $w = 1$ and two operators that can set v and w to 1. Applying the zero-cost operator $\mathit{Freeze}(v, 0)$ from partition $\text{SetO}_{v=0}$ is neither required nor corrupting an optimal plan (as it does not disable any operator from other partitions).

Theorem 2. *Let s be a solvable state. Then, for at least one operator partition $X \in P_s$, the induced strong semistubborn set T_s defined in Theorem 1 contains an operator that starts a strongly optimal plan in s. Hence T_s is a GSSS in the planning sense.*

Proof. Because s is solvable, there exists an operator o that starts a strongly optimal plan in s. As P_s contains *exactly* the applicable operators, one of these partitions contains o. □

We observe that only partitions inducing a GSSS in the planning sense are needed to find strongly optimal plans, whereas the others could ultimately be ignored. However, deciding if a partition induces a GSSS in the planning sense is computationally hard. Wolfe and Russell propose a heuristic criterion to select "good" partitions, which prefers partitions that resolve existing intentions. This in turn corresponds to selecting partitions inducing a GSSS in the planning sense. Hence, our analysis in particular sheds light on *what* the heuristic proposed in BIP computes and *why* it is reasonable.

4 CONCLUSION

BIP's operator partition pruning can be viewed as a stubborn set method: every applicable operator partition X induces a strong stubborn set in the Valmari sense with the same pruning power as X, and for every state, there must be at least one such partition that induces a generalized strong stubborn set in the planning sense.

Apart from the theoretical results obtained so far, our analysis also points us to interesting future research directions. In particular, as the "good" operator partitions are related to strong stubborn sets which are defined for arbitrary (non-unary) operators, it will be interesting to investigate if BIP can be generalized to arbitrary operators based on this insight – this question is considered as "most important" in Wolfe and Russell's future work description. We think that our analysis provides a promising starting point for this research goal.

ACKNOWLEDGEMENTS

This work was supported by the Swiss National Science Foundation as part of the project *Safe Pruning in Optimal State Space Search*.

REFERENCES

[1] Malte Helmert, 'The Fast Downward planning system', *JAIR*, **26**, 191–246, (2006).
[2] Silvan Sievers, Martin Wehrle, and Malte Helmert, 'Bounded intention planning revisited: Proof', Technical Report CS-2014-002, Universität Basel, Departement Mathematik und Informatik, (2014).
[3] Antti Valmari, 'Stubborn sets for reduced state space generation', in *Proc. APN 1989*, pp. 491–515, (1989).
[4] Martin Wehrle and Malte Helmert, 'About partial order reduction in planning and computer aided verification', in *Proc. ICAPS 2012*, (2012).
[5] Martin Wehrle and Malte Helmert, 'Efficient stubborn sets: Generalized algorithms and selection strategies', in *Proc. ICAPS 2014*, (2014).
[6] Martin Wehrle, Malte Helmert, Yusra Alkhazraji, and Robert Mattmüller, 'The relative pruning power of strong stubborn sets and expansion core', in *Proc. ICAPS 2013*, pp. 251–259, (2013).
[7] Jason Wolfe and Stuart J. Russell, 'Bounded intention planning', in *Proc. IJCAI 2011*, pp. 2039–2045, (2011).