Integrating Partial Order Reduction and Symmetry Elimination for Cost-Optimal Classical Planning

Martin Wehrle and Malte Helmert
University of Basel, Switzerland
martin.wehrle@unibas.ch
malte.helmert@unibas.ch

Alexander Shleyfman
Technion, Haifa, Israel
alesh@tx.technion.ac.il

Michael Katz
IBM Haifa Research Lab, Israel
katzm@il.ibm.com

Abstract

Pruning techniques based on partial order reduction and symmetry elimination have recently found increasing attention for optimal planning. Although these techniques appear to be rather different, they base their pruning decisions on similar ideas from a high level perspective. In this paper, we propose safe integrations of partial order reduction and symmetry elimination for cost-optimal classical planning. We show that previously proposed symmetry-based search algorithms can safely be applied with strong stubborn sets. In addition, we derive the notion of symmetrical strong stubborn sets as a more tightly integrated concept. Our experiments show the potential of our approaches.

1 Introduction

Heuristic search is a leading approach to optimally solving classical planning problems. However, for optimal planning, pure heuristic search based on $A^*$ is limited in the sense that even almost perfect heuristics can lead to exponentially large state spaces in typical planning problems [Helmert and Röger, 2008]. Hence, additional (and orthogonal) pruning techniques are desired to be able to efficiently scale to larger problems. Pruning techniques allow for considering only a subset of successors in every state (and thus potentially reduce the branching factor of the given planning task), while preserving completeness and optimality of optimal search algorithms. Recent approaches for optimal planning include strong stubborn sets [Alkhazraj et al., 2012; Wehrle et al., 2013; Wehrle and Helmert, 2014], symmetry elimination [Pochter et al., 2011; Domshlak et al., 2012; 2013], partition-based path pruning [Nissim et al., 2012], tunneling [Coles and Coles, 2010; Nissim et al., 2012], sleep sets [Wehrle and Helmert, 2012; Holte et al., 2015], move pruning [Holte and Burch, 2014], and commutativity pruning [Haslum and Geffner, 2000].

Recently, techniques based on partial order reduction and symmetry elimination have found particular attention in the planning community. From a technical point of view, these techniques appear to be rather different: While search-based planning algorithms with partial order reduction work on the original state space and prune “permutations” of plans in every state, algorithms based on symmetry elimination work on a modified state space (called the orbit space) that consists of equivalence classes of symmetrical states. Despite these differences, both approaches exploit the “equivalence” of permutations (variables and values vs. sequences of operators, respectively) for their pruning decisions. Originally, both partial order reduction and symmetry elimination stem from the area of model checking, where also combinations of them have been studied [Emerson et al., 1997; Bosnacki and Scheffer, 2015]. Overall, it naturally arises the question whether these techniques can be integrated safely and efficiently for the purpose of optimal planning as well.

In this paper, we propose safe integrations of partial order reduction based on strong stubborn sets and symmetry elimination for optimal planning. Our first approach applies strong stubborn sets “directly” as the basis for the orbit space computation obtained by symmetry elimination. We show that this combination is safe and optimality preserving when applied for planning. Our second approach, called symmetrical strong stubborn sets, provides a more tightly integrated concept based on restricting the original definition of strong stubborn sets to canonical (“symmetrical”) operators. We prove that symmetrical strong stubborn sets yield a safe state-based successor pruning function as introduced by Wehrle and Helmert [2014], which have the property that the pruning decision in a state $s$ is based solely on $s$. Such techniques can safely be applied in graph search algorithms like $A^*$ and are hence particularly attractive. We empirically investigate the performance of our approaches on the benchmarks from the international planning competitions.

2 Background

We consider $sas^+$ planning with a finite set of finite-domain state variables $\mathcal{V}$. Every variable $v \in \mathcal{V}$ has a finite domain $\text{dom}(v)$. A variable/value pair $\langle v,d \rangle$ for $v \in \mathcal{V}$ and $d \in \text{dom}(v)$ is called a fact. A partial state $s$ is defined as a function from variables $\text{vars}(s) \subseteq \mathcal{V}$ to values in the domains of $\text{vars}(s)$, whereas all variables in $\mathcal{V} \setminus \text{vars}(s)$ have an undefined value $\text{undef}$. We denote the value of $v$ in $s$ with $s[v]$ (including $s[v] = \text{undef}$ in case $v \in \mathcal{V} \setminus \text{vars}(s)$). A state is a partial state where all values are defined, i.e., with $\text{vars}(s) = \mathcal{V}$.

An operator $o$ is a tuple $\langle \text{pre}(o), \text{eff}(o) \rangle$, where $\text{pre}(o)$ and $\text{eff}(o)$ are partial states and denote the precondition and the
effect of \( o \), respectively. An operator \( o \) is applicable in a state \( s \) iff \( s[v] = \text{pre}(o)[v] \) for all \( v \in \text{vars(\text{pre}(o))} \). If \( o \) is applicable in \( s \), the successor state \( o(s) \) of \( s \) is obtained from \( s \) by setting the values of variables in \( \text{vars(\text{eff}(o))} \) to their values in \( \text{eff}(o) \), and leaving the remaining variable values unchanged. We denote the set of applicable operators in \( s \) with \( \text{app}(s) \). Furthermore, we say that an operator \( o \) is an achiever of a fact \( (v,d) \) if \( \text{eff}(o)[v] = d \).

A \( \text{SAS}^+ \) planning task is a tuple \( \Pi = \langle V, O, s_0, s_*, C \rangle \), consisting of a finite set of finite-domain state variables \( V \), a finite set of operators \( O \), an initial state \( s_0 \), and a partial goal state \( s_* \). In addition, we are given a cost function \( C \) that assigns a non-negative cost value \( C(o) \) to each operator \( o \in O \). We denote the set of all states of \( \Pi \) with \( S \). The state transition graph \( T_{\Pi} = \langle S, E \rangle \) of \( \Pi \) is a directed graph whose set of vertices is the set of states \( S \), and there is an edge \( (s, s', o) \in E \) from \( s \) to \( s' \) labeled with \( o \) iff \( o \) is applicable in \( s \) and \( s' = o(s) \). A plan in \( S \) is a sequence of operators \( \sigma = o_1 \ldots o_n \) such that \( o \) is sequentially applicable in \( s \) and leads to a state that complies with \( s_* \). A state \( s \) is called solvable if there is a plan in \( s \). A plan \( \pi \) is optimal if \( \pi \) is optimal and contains a minimal number of zero-cost operators. In the following, we focus on finding optimal plans based on \( A^* \) search [Hart et al., 1968].

2.1 Strong Stubborn Sets

Strong stubborn sets yield safe successor pruning functions as defined by Wehrle and Helmert [2014]: A successor pruning function for a planning task \( \Pi \) is a function \( f : S \rightarrow O^\Pi \) that maps states \( s \) to subsets \( \text{ops}(s) \) of applicable operators in \( s \), i.e., \( \text{ops}(s) \subseteq \text{app}(s) \). A successor pruning function is safe if in a modified state transition graph where only the operators in \( \text{ops}(s) \) may be applied in all states \( S \), the original solution costs remain the same. As a sufficient criterion, Wehrle and Helmert show that if for all solvable non-goal states \( s \), \( \text{ops}(s) \) contains at least one operator that starts a strongly optimal plan, then the corresponding successor pruning function is safe. In the following, we call pruning functions that satisfy this criterion strongly safe.

Recently, strong stubborn sets have been considered in different generalizations. In the following, we introduce a variant that is general enough for the purpose of this paper, based on a state-dependent notion of interference [Wehrle and Helmert, 2014]. For this, we need more terminology. Firstly, we say that two operands \( o, o' \) interfere in a state \( s \) iff both \( o \) and \( o' \) are applicable in \( s \), and at least one of the following conditions holds: \( o \) disables \( o' \) (i.e., \( o' \) is not applicable in \( o(s) \)), or vice versa, or \( o(\text{ops}(s)) \) and \( o(\text{ops}(s)) \) are both defined, but \( o(\text{ops}(s)) \neq o'(\text{ops}(s)) \). Secondly, for a planning task \( \Pi = \langle V, O, s_0, s_*, C \rangle \), for a state \( s \) and an operator \( o \in O \), a necessary enabling set for \( o \) and \( s \) is a set of operators \( N \) such that for all plans \( \pi \) that start in \( s \) and include \( o \), some \( o' \in N \) occurs in \( \pi \) before the first occurrence of \( o \). Furthermore, a disjunctive action landmark \( L \) in a state \( s \) is a set of operators such that every plan in \( s \) includes at least one operator of \( L \).

Definition 1 (strong stubborn set) Let \( s \) be a solvable non-goal state in planning task \( \Pi = \langle V, O, s_0, s_*, C \rangle \). A set of operators \( T(s) \) is a strong stubborn set in \( s \) if

1. \( T(s) \) contains a disjunctive action landmark in \( s \),
2. for every \( o \in T(s) \) not applicable in \( s \), \( T(s) \) contains a necessary enabling set for \( o \) and \( s \), and
3. for every \( o \in T(s) \) applicable in \( s \), \( T(s) \) contains all operators that interfere with \( o \) in \( s \).

2.2 Symmetry Elimination

In contrast to strong stubborn sets, symmetry elimination considers equivalence classes of symmetrical states, and allows for using representative states of each equivalence class. Recently, Shleifman et al. [2015] introduced the notion of structural symmetries, which capture previously proposed concepts of symmetry for classical planning. In a nutshell, structural symmetries directly work on the factored representation of a given planning task \( \Pi \). They map operators to operators, and variable/value pairs to variable/value pairs in such way that the induced mapping on the state transition graph \( T_{\Pi} \) is an automorphism of \( T_{\Pi} \). More formally, structural symmetries for \( \text{SAS}^+ \) planning tasks are defined as follows.

Definition 2 (structural symmetry) For a planning task \( \Pi = \langle V, O, s_0, s_*, C \rangle \), let \( F \) be the set of \( \Pi \)'s facts, i.e., pairs \((v,d)\) with \( v \in V \) and \( d \in \text{dom}(v) \). A structural symmetry for \( \Pi \) is a permutation \( \sigma : F \cup O \rightarrow F \cup O \) where

1. \( \sigma(\text{FV}) = \text{FV} \), where \( \text{FV} := \{ (v,d) \mid d \in \text{dom}(v) \} \)
2. \( \sigma(\text{O}) = \text{O} \) such that for \( o \in \text{O} \), \( \sigma(\text{pre}(o)) = \text{pre}(\sigma(o)) \), \( \sigma(\text{eff}(o)) = \text{eff}(\sigma(o)) \), and \( C(\sigma(o)) = C(o) \).
3. \( \sigma(s_*) = s_* \),

where \( \sigma((x_1, \ldots, x_n)) := (\sigma(x_1), \ldots, \sigma(x_n)) \), and for a partial state \( s, s' := \sigma(s) \) is the partial state obtained from \( s \) such that for all \((v,d)\) with \( v \in \text{vars}(s) \) and \( d \in \text{dom}(v) \), \( \sigma((v,d)) = (\sigma(v), \sigma(d)) \) and \( s'[\sigma(v)] = \sigma(d) \).

For a planning task \( \Pi \) with states \( S \), a set of structural symmetries \( \Sigma \) induces a group \( \Gamma \) and an equivalence relation \( \sim_{\Gamma} \) on \( S \), where \( s \sim_{\Gamma} s' \) iff there is \( \sigma \in \Gamma \) such that \( \sigma(s) = s' \).

For a state \( s \), pruning algorithms based on symmetry elimination only consider the equivalence classes of the successor states of \( s \) instead of all successor states, and only keep one representative element of these classes. In this sense, \( A^* \) with symmetry elimination applies all operators in \( s \), but prunes some of the resulting successor states. The resulting reduced state transition graph is guaranteed to still contain an optimal plan in \( s \). To achieve this, Domshlak et al. [2012] have introduced a variant of the \( A^* \) algorithm (called DKS) that performs duplicate elimination based on canonical states. For that, the DKS algorithm maintains an additional (i.e., the canonical) state for each search node, leading to an increased memory consumption. To overcome this problem, a further variant of \( A^* \) called orbit space search (OSS) has been introduced [Domshlak et al., 2015]. Orbit space search directly performs the search in the orbit space which is induced by canonical representatives. In other words, orbit space search searches the state transition graph induced by canonical representatives for each encountered state, i.e., every state is mapped to a corresponding representative.
3 Strong Stubborn Sets + Orbit Space Search

Strong stubborn sets and symmetries are orthogonal concepts in terms of pruning power. For space reasons, we refer to a technical report showing examples where strong stubborn sets can prune more than symmetries, and vice versa [Wehrle et al., 2015]. Following previous work in model checking [Emerson et al., 1997; Bosnacki and Scheffer, 2015], our first approach “directly” integrates strong stubborn sets and symmetry elimination for optimal planning: In every state \(s\), instead of computing canonical states for all successor states of \(s\), the computation of canonical states is restricted to those successors that result from a strong stubborn set in \(s\). The proposed modification is both completeness and optimality preserving. In the following, we prove the more general result that both DKS and OSS can be safely combined with strongly safe successor pruning functions.

Theorem 1 Restricting the successor generation of the DKS and OSS algorithms with a strongly safe successor pruning function yields complete and optimal planning algorithms.

Proof: Let \(f\) be some strongly safe successor pruning function. For each expanded state \(s\), we have that there exists a plan for \(s\), then there exists a strongly optimal plan \(\pi_s = o_1, \ldots, o_n\) for \(s\) such that \(o_1 \in f(s)\). Then \(s_1 = o_1(s)\) is one of the successors generated using \(f\). DKS will prune \(s_1\) only if some other state \(s'_1\) with the same canonical representative was already generated by DKS. OSS will generate the canonical representative \(s'_1\) of \(s_1\) instead of \(s_1\). In both cases, at least one representative state in each equivalence class remains eligible for expansion at each time. For each plan from \(s_1\), there exists a plan of the same cost from \(s'_1\) and there exists one from \(s_1\) as well. Thus, as DKS and OSS are complete and optimal planning algorithms, the claim follows with a simple induction over the length of the strongly optimal plan. □

3.1 Experimental Evaluation

We have implemented the resulting search algorithms in the Fast Downward planner [Helmer, 2006] on top of the strong stubborn set implementation using full envelopes, mutex-based inference, and static FD ordering [Wehrle and Helmer, 2014], and on top of the orbit space search implementation [Domshlak et al., 2015]. Our experiments were performed on machines with Intel Xeon E5-2660 CPUs running at 2.2 GHz. The time and memory bounds used per run were 30 minutes and 2 GB, respectively. All configurations used the LM-Cut heuristic [Helmer and Domshlak, 2009], the state-of-the-art heuristic for optimal planning, on all optimal STRIPS IPC benchmarks up to 2011 that are supported by LM-Cut (44 domains, 1396 instances). The results are given in Figure 1.

We observe that the strengths of strong stubborn sets and symmetry elimination are rather orthogonal: Our integration, called \(o/s\) in Fig. 1, mostly achieves at least the same coverage (i.e., number of solved problems) as the maximum coverage of the previous approaches. In particular, this is the case in domains where the coverage of the previous approaches is very different (e.g., in Gripper and Parcprinter, respectively). Overall, our approach solves 859 out of 1396 problems, which is particularly remarkable due to the usual exponential complexity increase in the size of the problems. Considering the number of expanded states, we observe a more fine grained picture: While still in most domains the number of expansions with \(o/s\) is at most as high as the minimum of the previous approaches, in 19 out of these the number of expansions is strictly lower than the minimum. Although the difference is sometimes moderate, it shows that there exist synergy effects which could further be exploited by future benchmark problems. In addition, in the Mystery, Rovers and Satellite domains, this synergy effect is already strong enough to yield the uniquely highest coverage.

4 Symmetrical Strong Stubborn Sets

The concept of symmetrical strong stubborn sets integrates strong stubborn sets and symmetries more tightly. It is based on restricting strong stubborn sets to canonical representatives of equivalence classes of symmetrical operators.

4.1 Canonical Operators

We will derive canonical operators by applying symmetry-based reasoning to operators. This concept will provide a
strongly safe successor pruning function in its own right, and
will form the basis for symmetrical strong stubborn sets.
While it might appear obvious to achieve these objectives for
a given structural symmetry group (i.e., by just considering
the representative operators of each equivalence class of op-
erators induced by the given symmetries), care must be taken
with the details: structural symmetries \( \sigma \) that do not stabi-
lize the current state \( s \) (i.e., \( \sigma(s) \neq s \)) are not guaranteed to
yield safe successor pruning functions. For brevity, we again
refer to a technical report for an example showing that non-
stabilizing symmetries do not yield strongly safe successor
pruning functions in general [Wehrle et al., 2015].

In the following, we make these ideas more precise. Let
\( \Pi = (\mathcal{V}, \mathcal{O}, s_0, s_*, C) \) be a planning task, \( \Sigma \) be a set of struc-
tural symmetries of \( \Pi \), and \( s \) be a state of \( \Pi \). Let \( \Sigma_s \subseteq \Sigma \)
be the set of structural symmetries that stabilize \( s \) (that is,
\( \sigma(s) = s \) for all \( \sigma \in \Sigma_s \)). Let \( \Gamma_s \) be a group induced by \( \Sigma_s \)
and let \( \sim_s \) be the equivalence relation over the operator set
\( \mathcal{O} \) induced by \( \Gamma_s \). The relation \( \sim_s \) defines a partitioning
of the operator set \( \mathcal{O} \) into equivalence classes. Each equivalence
class is identified with one of the operators from the class,
which is chosen to be the canonical operator for that equiva-
lence class. Slightly abusing the notation, the canonical oper-
ator for the operator \( o \in \mathcal{O} \) is denoted by \( [o]_s \). The mapping
\( \mathcal{C}_L : \mathcal{O} \mapsto \mathcal{O} \) defined by \( \mathcal{C}_L(o) := [o]_s \) is called the canoni-
cal operator labeling in \( s \). We denote the induced successor
pruning function \( \text{sop}(s) := \{ \mathcal{C}_L(o) \mid o \in \text{app}(s) \} \) as sym-
metrical operator pruning. Symmetrical operator pruning is
strongly safe, and more generally, it can safely be combined
with any strongly safe successor pruning function.

**Theorem 2** Let \( f \) be a strongly safe successor pruning func-
tion, and let \( \text{ops} \) be the successor pruning function defined as
\( \text{ops}(s) := \{ \mathcal{C}_L(o) \mid o \in f(s) \} \). Then \( \text{ops} \) is strongly safe.

**Proof:** Let \( \pi = o_1, \ldots, o_n \) be a strongly optimal plan for \( s \)
such that \( o_1 \in f(s) \). Such a plan exists, since \( f \) is strongly safe.
Let \( \sigma \) be some structural symmetry stabilizing \( s \) such
that \( \sigma(o_1) = \mathcal{C}_L(o_1) \). Then \( \pi' = \sigma(\pi) = \sigma(o_1), \ldots, \sigma(o_n) \)
is a strongly optimal plan for \( s = \sigma(s) \), starting with an oper-
ator in \( \text{ops}(s) \). Thus \( \text{ops} \) is strongly safe. \( \square \)

Computing canonical operator labelings for a given state is
polynomial time. Let \( o_1, \ldots, o_k \) be the operators of a plan-
ing task, with \( i \) being the index of \( o_i \). For a state \( s \), a canoni-
cal operator labeling \( \mathcal{C}_L \) for \( s \) is computed as follows.

1. In a pre-search phase, we compute for each gener-
ator \( \sigma \) a canonical operator labeling \( \mathcal{C}_L \) for \( \sigma \) as fol-
ows: Starting with \( \mathcal{C}_L(o_i) = i \), we iteratively set
\( \mathcal{C}_L(o) := \min(\mathcal{C}_L(o), \mathcal{C}_L(\sigma(o))) \) for each operator
\( o \) until a fixed-point is reached.

2. During search, for a set of generators stabilizing \( s \), we
compute \( \mathcal{C}_L \) for \( s \) by starting with an identity labeling,
and continue the following procedure until a fixed-point
in \( \mathcal{C}_L \) is reached: For each stabilizing generator \( \sigma \), for
each operator \( o, \mathcal{C}_L(o) := \min(\mathcal{C}_L(\sigma(o)), \mathcal{C}_L(o)) \).

Since the indices only reduce, the number of iterations is
bounded by \( O(k^2) \), with each iteration being linear in \( k \)
and the number of symmetry group generators.

Overall, Theorem 2 allows us to safely use symmetrical oper-
ator pruning in polynomial time within \( A^* \), and more gener-
ally, it allows us to use it on top of any strongly safe successor
pruning functions. In addition, Theorem 2 will serve as an in-
grident for integrating strong stubborn sets and symmetries.

### 4.2 Symmetrical Strong Stubborn Sets

As outlined, symmetrical strong stubborn sets restrict the de-
финition of strong stubborn sets to canonical operators.

**Definition 3 (symmetrical strong stubborn set)** Let \( \Pi = (\mathcal{V}, \mathcal{O}, s_0, s_*, C) \) be a planning task, let \( s \) be a state of \( \Pi \), and
\( \mathcal{C}_L \) be the canonical operator labeling in \( s \). A symmetrical
strong stubborn set (SSSS) in \( s \) is a set of operators \( H \subseteq \mathcal{O} \)
with the following properties. If \( s \) is an unsolvable or goal
state, every set \( H \subseteq \mathcal{O} \) is a SSSS. If \( s \) is a solvable non-goal
state, then \( H \) satisfies the following constraints:

1. \( H \) contains the canonical operators of a disjunctive ac-
ction landmark \( L \) in \( s \), i.e., \( \{ \mathcal{C}_L(o) \mid o \in L \} \subseteq H \).

2. for every \( o \in H \) not applicable in \( s \), \( H \) contains the canoni-
cal operator labeling of a necessary enabling set \( N \) for \( o \) and \( s \), i.e., \( \{ \mathcal{C}_L(o) \mid o \in N \} \subseteq H \), and

3. for every \( o \in H \) applicable in \( s \), \( H \) contains the canoni-
cal operator labeling of all operators \( o' \) that interfere
with \( o \), i.e., \( \{ \mathcal{C}_L(o') \mid o \text{ interferes with } o' \text{ in } s \} \subseteq H \).

Symmetrical strong stubborn sets yield a safe successor pruning
function.

**Theorem 3** Let \( \text{ops} \) be a successor pruning function defined as
\( \text{ops}(s) = H(s) \cap \text{app}(s) \), where \( H(s) \) is a symmetrical
strong stubborn set in \( s \). Then \( \text{ops} \) is strongly safe.

**Proof:** Our proof is based on the proof of Theorem 1 by
Wehrle and Helmut [2014]. Let \( s \) be a state and \( H \) be a SSSS
in \( s \). We show that if \( s \) is a solvable non-goal state, then \( H \)
contains an operator which is the first operator in a strongly
optimal plan for \( s \). The claim then follows with Proposition 1
of Wehrle and Helmut [2014]. In the following, we refer to the
three conditions of Def. 3 as C1–C3.

Let \( \pi = o_1, \ldots, o_n \) be a strongly optimal plan for \( s \)
such that \( \mathcal{C}_L(o_i) \in H \) for some \( i \in \{1, \ldots, n\} \). Such a plan must
exist because of C1. Let \( k \in \{1, \ldots, n\} \) be the minimal index
for which \( \mathcal{C}_L(o_k) \in H \) and let \( \sigma_k \) be a structural
symmetry that stabilizes \( s \) such that \( \mathcal{C}_L(o_i) = \sigma_k(o_i) \).

We show by contradiction that \( \sigma_k(o_i) \) is applicable in \( s \).
Assume it is not applicable. Since \( \sigma_k(o_i) \in H \), C2 guarantees that
\( H \) contains the canonical operator labeling of a necessary en-
abling set for \( \sigma_k(o_i) \). Let \( \pi_k = \sigma_k(\pi) = \sigma_k(o_1), \ldots, \sigma_k(o_n) \).
\( \pi_k \) is a strongly optimal plan for \( s \), since it is a mapping of
\( \pi \) with a structural symmetry that stabilizes \( s \). Therefore,
a necessary enabling set for \( \sigma_k(o_i) \) will include the canonical oper-
ator labeling of some operator \( \sigma_k(o_i) \) for \( i < k \). Since
\( \mathcal{C}_L(\sigma_k(o_i)) = \mathcal{C}_L(o_i) \), according to C2, \( H \) must contain
the canonical operator labeling of \( o_i \), contradicting the mini-
mality of \( k \). It follows that \( \sigma_k(o_i) \) is applicable in \( s \).

Let \( s^0, \ldots, s^n \) be the sequence of states visited by \( \pi_k \):
\( s^0 = s \) and \( s^i = \sigma_k(o_i)(s^{i-1}) \) for all \( i \in \{1, \ldots, n\} \). It
follows that $o^*_C$ does not interfere with any of the operators $\sigma_k(o_1), \ldots, \sigma_k(o_{k-1})$ in any of the states $s^j$: if it did, then from C3 (with $o = o^*_C$), the canonical operator labeling of the interfering operators would be contained in $H$, together with $CL_s(\sigma_k(o_1)) = CL_s(o_1)$ contradicting the minimality of $k$.

The remainder of the proof, showing that if $o^*_C$ is not already the first operator in $\pi_k$, it can be shifted to the front of $\pi_k$, is exactly as in Wehrle and Helmert (2014). □

Symmetrical strong stubborn sets generalize symmetrical operator pruning and stubborn sets in the following sense.

**Theorem 4** Let $G$ be a strong stubborn set in $s$. Then $H = \{ CL_s(o) \mid o \in G \}$ is a symmetrical strong stubborn set in $s$.

**Proof:** It is clear that C1 will hold for $H$. To see that C2 holds, let $o \in G$ be some non-applicable operator and let $N \subseteq G$ be its necessary enabling set. Let $\sigma$ be some structural symmetry stabilizing $s$ such that $\sigma(o) = CL_s(o)$. Then $N' = \{ \sigma(o') \mid o' \in N \}$ is a necessary enabling set for $CL_s(o)$. Note that for all actions $o' \in N$ we have $CL_s(o') = CL_s(\sigma(o'))$ and thus $N' = \{ CL_s(o') \mid o' \in N \}$ is the canonical operator labeling of a necessary enabling set for $CL_s(o)$. To see that C3 holds as well, let $o \in G$ be some applicable operator and let $o'$ interfere with $o$ in $s$. Let $\sigma$ be some structural symmetry that stabilizes $s$ such that $\sigma(o) = CL_s(o)$. Then $\sigma(o)$ interferes with $\sigma(o')$ in $s = \sigma(s)$. Thus, $CL_s(\sigma(o')) = CL_s(o') \in H$. □

We remark that although Theorem 4 shows a theoretical dominance result, the choice of the algorithms in practical implementations does not necessarily guarantee the dominance in terms of state explorations. However, in our experiments, the latter is established in almost all domains (we will come back to this point in the next section).

In addition, symmetrical strong stubborn sets offer the potential to prune more than the combination of strong stubborn sets with orbit space search according to Theorem 1, and than the combination of strong stubborn sets and symmetrical operator pruning according to Theorem 2. Intuitively, this is the case because symmetrical strong stubborn sets recognize (and can exploit) symmetries also for inapplicable operators.

**Example 1** Let $\Pi_1$ be a planning task with binary variables $V = \{a, b, c, d, g\}$ and uniform-cost operators $O = \{o_1, o_2, o_3, o_4, o_5\}$ with

- $pre(o_1) = \{(a, 1), (b, 1)\}$, $eff(o_1) = \{(g, 1)\}$
- $pre(o_2) = \{(b, 1), (c, 1)\}$, $eff(o_2) = \{(g, 1)\}$
- $pre(o_3) = \emptyset$, $eff(o_3) = \{(a, 1)\}$
- $pre(o_4) = \emptyset$, $eff(o_4) = \{(c, 1)\}$
- $pre(o_5) = \{(d, 1)\}$, $eff(o_5) = \{(b, 1)\}$.

Let $s_0 = \{(a, 0), (b, 0), (c, 0), (d, 1), (g, 0)\}$ and $s_* = \{(g, 1)\}$. We observe that there is a structural symmetry $\sigma$ that maps operator $o_1$ to $o_2$ and $o_3$ to $o_4$, and vice versa, and stabilizes the initial state (by mapping $a = 1$ and $c = 1$ to each other). Without loss of generality, assume that the canonical operator for both $o_1$ and $o_2$ is $o_1$ and for both $o_3$ and $o_4$ is $o_3$.

- Consider the combination of strong stubborn sets with orbit space search (oss) based on Theorem 1. Strong stubborn sets in $s_0$ can be obtained according to the following procedure: $\{o_1, o_2\}$ is a disjunctive action landmark (consisting of the operators that set the goal variable). A necessary enabling set for $o_1$ is $\{o_3\}$, a corresponding set for $o_2$ is $\{o_5\}$, resulting in the strong stubborn set $\{o_1, o_2, o_3, o_5\}$. Out of this set, $o_2$ and $o_5$ are applicable. Furthermore, as previously proposed algorithms for symmetry detection base their computation on a syntactic planning task description [Pochter et al., 2011], the successor of $o_0$ under $o_5$ will not be recognized as symmetrical to $s_0$'s other successors (like the successor under $o_3$ in particular). Thus orbit space search with strong stubborn sets will classify $s_0$'s successors under $o_3$ and $o_5$ in different orbits.

- Consider the successor pruning function obtained by the combination of strong stubborn sets and symmetrical operator pruning based on Theorem 2: Starting with the strong stubborn set according to the description in the bullet above, $o_3$ and $o_5$ out of this set are applicable, but not symmetrical, hence no further reductions are obtained. Thus two successor states are generated.

In contrast, symmetrical strong stubborn sets only produce one successor state in $s_0$ because of the restriction to canonical operators: Only $o_1$ is included due to C1 (compared to $o_1$ and $o_2$ for the other methods), and $o_5$ is included due to C2, resulting in a set that only contains one applicable operator $o_3$. Intuitively, symmetrical strong stubborn sets can achieve more pruning as they already recognize symmetries in “intermediate” steps during the fixed-point computation.

The example shows that symmetrical strong stubborn sets can further increase the pruning power under reasonable practical design choices w.r.t. their computation (i.e., using the achievers of an unsatisfied goal fact as disjunctive action landmark, and using the achievers of the first unsatisfied precondition fact as necessary enabling sets). We remark that selecting the achievers of a first unsatisfied precondition fact according to a particular ordering for computing necessary enabling sets (e.g., selecting the achievers of $(b, 1)$ for $o_2$ in the initial state) can be viewed as tie-breaking. However, this is a way strong stubborn sets have recently been successfully implemented, e.g., by Alkhazraji et al. [2012].

### 4.3 Experimental Evaluation

We have implemented and evaluated symmetrical strong stubborn sets in the same setting as in the previous experimental section. The coverage results are given in Fig. 2. We compare symmetrical operator pruning (called sop), symmetrical strong stubborn sets (ssss), and the combination of standard strong stubborn sets and symmetrical operator pruning according to Theorem 2 (ss/s) with standard strong stubborn sets (sss) within $A^*$ and orbit space search.

We observe that symmetrical strong stubborn sets yield a slightly higher coverage than strong stubborn sets within $A^*$, and lowers the coverage within OSS. One reason for the latter is the overhead for computing canonical operators. The relative coverage difference of $A^*$ and OSS (slightly improved
Firstly, we compare symmetrical strong stubborn sets to strong stubborn sets. For $A^*$, Fig. 3 shows that symmetrical strong stubborn sets can further reduce the number of expansions (by several orders of magnitude in some cases). In contrast, for OSS, only slightly fewer states are expanded (by several orders of magnitude in some cases). This difference in state expansions in turn explains the relative coverage difference of these configurations as discussed above.

Secondly, symmetrical strong stubborn sets show promising search behavior compared to symmetrical operator pruning: for both $A^*$ and OSS, symmetrical strong stubborn sets expand (sometimes significantly) fewer states than symmetrical operator pruning. The resulting scatterplots (which are not shown again for space reasons) look similar to the

![Figure 3: Expansions (without last $f$ layer) for $A^*$ + LM-Cut: strong stubborn sets vs. symmetrical strong stubborn sets](image)

plot in Fig. 3. Thirdly, symmetrical strong stubborn sets compared to the combination of strong stubborn sets and symmetrical operator pruning yields almost no additional pruning in almost all domains (for both $A^*$ and OSS). Apparently, in practice, OSS already mostly captures the additional pruning power offered by symmetrical strong stubborn sets.

Overall, the experiments show that the generalization result of Theorem 4 often carries over to practice. However, the additional pruning offered by symmetrical strong stubborn sets is exploited in terms of coverage by actual implementations only partly. Nevertheless, for $A^*$, there are problems where the additional pruning power is significant and pays off.

5 Related Work

Combinations of partial order reduction and symmetry elimination have already been (and still are) studied in the area of computer aided verification [Emerson et al., 1997; Bosnacki and Scheffer, 2015]. Both Emerson et al. and Bosnacki and Scheffer consider partial order reduction based on ample sets. While Emerson et al. require considering unique canonical states within orbit space search, Bosnacki and Scheffer extend this theory by also allowing multiple representatives. Like the latter, we allow using multiple representatives: the approach by Bosnacki and Scheffer corresponds to our integration of strong stubborn sets and orbit space search for goal reachability, which we have additionally shown to be optimality preserving. Symmetrical strong stubborn sets further extend these concepts for goal reachability. In addition, we have empirically shown these approaches to be useful on a large number of planning benchmarks.

6 Conclusions

We have proposed two integration approaches of partial order reduction and symmetry elimination for planning, and proved them to be completeness and optimality preserving.
Our experiments show that the most direct integration is already most powerful in terms of coverage: restricting the orbit space search to states generated by strong stubborn sets often inherits the strengths of both partial order reduction and symmetry elimination, and significantly increases the number of solved problems in the standard benchmark suite from the international planning competitions. Furthermore, our concept of symmetrical strong stubborn sets offers additional pruning power compared to previous approaches, which is partly exploited by our current implementations. For the future, it will be interesting to further investigate the questions if more powerful integrations of partial order reduction and symmetry elimination can be derived, and to which extent the pruning power can be carried over to practice.

Acknowledgments
This work was supported by the Swiss National Science Foundation (SNSF) as part of the project “Automated Reformulation and Pruning in Factored State Spaces (ARAP)”, by the Technion-Microsoft Electronic-Commerce Research Center, and by ISF grant 1045/12.

References