Heuristics and Symmetries in Classical Planning

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

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

Malte Helmert and **Silvan Sievers** and **Martin Wehrle**
University of Basel, Switzerland
{malte.helmert,silvan.sievers,martin.wehrle}@unibas.ch

Abstract

Heuristic search is a state-of-the-art approach to classical planning. Several heuristic families were developed over the years to automatically estimate goal distance information from problem descriptions. Orthogonally to the development of better heuristics, recent years have seen an increasing interest in symmetry-based state space pruning techniques that aim at reducing the search effort. However, little work has dealt with how the heuristics behave under symmetries. We investigate the symmetry properties of existing heuristics and reveal that many of them are invariant under symmetries.

Introduction

Many current algorithms for classical planning are based on heuristic search in the problem state space (e. g., Hoffmann and Nebel 2001; Gerevini and Serina 2002; Helmert 2006; Richter and Westphal 2010). These planners use automatically derived heuristic functions to guide the search towards goal states. Their heuristics are often classified into four families: *abstractions* (e. g., Culberson and Schaeffer 1998; Edelkamp 2001; Helmert et al. 2014; Katz and Domshlak 2010), *delete relaxations* (e. g., Bonet and Geffner 2001; Hoffmann and Nebel 2001; Keyder and Geffner 2008; Katz, Hoffmann, and Domshlak 2013), *critical paths* (Haslum and Geffner 2000), and *landmarks* (e. g., Richter, Helmert, and Westphal 2008; Karpas and Domshlak 2009; Helmert and Domshlak 2009; Keyder, Richter, and Helmert 2010).

Besides heuristics, several state pruning methods have been proposed to alleviate the state explosion problem. One such method is *symmetry-based state pruning*, which has recently seen increasing interest in domain-independent planning (Pochter, Zohar, and Rosenschein 2011; Domshlak, Katz, and Shleyfman 2012; 2013). Symmetry pruning computes equivalence classes of symmetric states, and only explores one representative state per equivalence class.

Previous work on symmetry pruning in classical planning has, for the most part, not considered the question how these symmetries interact with heuristics. An exception is the work of Domshlak et al. (2013), which shows how information about symmetries can be used to enrich knowledge about landmarks that need to be achieved from a given search node.

In this paper, we systematically investigate the interaction of symmetries and heuristics for classical planning. In our main contribution, we analyze influential heuristics from the planning literature to see if they are *invariant under symmetry* in the sense that, given two symmetric states, they are guaranteed to compute the same estimate.

Invariance under symmetry gives us some reassurance that a heuristic captures global structural aspects of the problem: (blind) search trees rooted at symmetric states are isomorphic, and hence an unbiased heuristic might be expected to treat them identically. However, the state of the art in the classical heuristic search literature contains many examples of estimators that are *not* invariant under symmetry, and hence so-called *symmetric lookups* are often performed to compute heuristic information not just for the currently considered search state, but also for other representatives of its equivalence class (e. g., Culberson and Schaeffer 1998; Korf and Felner 2002; Felner et al. 2011). Such symmetric lookups are of course only useful for heuristics which are not invariant under symmetry.

As a second contribution, we introduce *structural symmetries* for classical planning tasks. Compared to other notions of symmetry suggested in the recent planning literature (e. g., Pochter, Zohar, and Rosenschein 2011), structural symmetries are directly defined in terms of straight-forward invariance properties on the original factored representation of a planning task. They are hence easier to understand and easier to reason about than previous notions of symmetry.

Background

We consider planning tasks $\Pi = \langle P, O, I, G, C \rangle$ in the (propositional) STRIPS formalism extended with operator costs. In such a task, P is a set of Boolean *propositions*. Each subset $s \subseteq P$ is called a *state*, and $S = 2^P$ is the *state space* of Π. The state I is the *initial state* of Π. The goal $G \subseteq P$ is a set of propositions, where a state s is a *goal state* if $G \subseteq s$. The set O is a finite set of *operators*. Each operator $o \in O$ has an associated set of *preconditions* $pre(o) \subseteq P$, *add* effects $add(o) \subseteq P$ and *delete* effects $del(o) \subseteq P$, and $C : O \to \mathbb{R}^{0+}$ is a non-negative *operator cost* function.

The semantics of STRIPS planning is as follows. An operator o is *applicable* in the state s if $pre(o) \subseteq s$. Applying o in s results in the state $s[\![o]\!] := (s \setminus del(o)) \cup add(o)$. The *transition graph* $\mathcal{T}_\Pi = \langle S, E \rangle$ of Π is the edge-labeled digraph over S which contains an edge $\langle s, s[\![o]\!]; o\rangle$ from s to $s[\![o]\!]$ labeled with o whenever $o \in O$ is applicable in state s. A sequence of operators $\pi = \langle o_1, \dots, o_k\rangle$ is *applicable* in s if the transition graph contains a path with label sequence π starting from s_0. If it exists, such a path is uniquely defined, and its end state is denoted by $s[\![\pi]\!]$. An applicable operator sequence is a *plan* for s if $s[\![\pi]\!]$ is a goal state. Its *cost* is the cumulative cost of operators in the sequence: $C(\pi) = \sum_{i=1}^{k} C(o_i)$. A plan for s with minimal cost is called *optimal*. The *perfect heuristic* for s, denoted by $h^*(s)$, or $h^*(s, \Pi)$ if the planning task is not clear from context, is the cost of an optimal plan for s. The objective of (optimal) planning is to find an (optimal) plan for I.

Symmetries of the State Transition Graph

A *symmetry* of a transition graph $\mathcal{T}_\Pi = \langle S, E\rangle$ with operators O is a permutation σ of $S \cup O$ mapping states to states and operators to operators such that
– $\langle s, s'; o\rangle \in E$ iff $\langle\sigma(s), \sigma(s'); \sigma(o)\rangle \in E$,
– $C(\sigma(o)) = C(o)$, and
– s is a goal state iff $\sigma(s)$ is a goal state
for all states s, s' and operators o. Symmetries are also called *(goal-stable) automorphisms*. They are closed under composition and inverse, forming the *automorphism group* *Aut*$(\mathcal{T}_\Pi)$ of the transition graph. Each subgroup Γ of symmetries induces an equivalence relation $\sim_\Gamma$ on states S: $s \sim_\Gamma s'$ iff $\sigma(s) = s'$ for some $\sigma \in \Gamma$. States in the same equivalence class are called *symmetric*. Computing (a compact representation of) *Aut*(G) for a graph G is not known to be polynomial-time, but backtracking techniques are surprisingly effective in finding substantial subgroups of *Aut*(G).

For notational convenience, throughout the paper we extend permutations σ over a carrier set X to sequences over X ($\sigma(\langle x_1, \dots, x_n\rangle) := \langle\sigma(x_1), \dots, \sigma(x_n)\rangle$) and subsets of X ($\sigma(\{x_1, \dots, x_n\}) := \{\sigma(x_1), \dots, \sigma(x_n)\}$).

The following (immediate) result is the formal basis for exploiting symmetries for planning:

Theorem 1 *Let Π be a planning task, let s be one of its states, let π be a sequence of operators of Π, and let σ be a symmetry of $\mathcal{T}_\Pi$. Then π is a plan for s iff $\sigma(\pi)$ is a plan for $\sigma(s)$, and the two plans have the same cost.*

A direct corollary is that $h^*(s) = h^*(\sigma(s))$ for all symmetries σ.

Symmetries from Problem Description Graphs

Pruning state spaces by reasoning about symmetries has been adopted in model checking (e. g., Emerson and Sistla 1996), constraint satisfaction (e. g., Puget 1993), and planning (e. g., Rintanen 2003; Fox and Long 1999; 2002; Pochter, Zohar, and Rosenschein 2011; Domshlak, Katz, and Shleyfman 2012; 2013).

As the state transition graph $\mathcal{T}_\Pi$ of a planning task Π is usually too large to be given explicitly, symmetries must be inferred from a compact description. Pochter et al. introduced a method for deducing some symmetries of the planning task from automorphisms of a certain graphical structure, the *problem description graph* (PDG) of Π. Later, Domshlak et al. (2012) slightly modified the definition, mainly to add support for general-cost actions. As observed by Pochter et al., every automorphism of the PDG of Π induces an automorphism of $\mathcal{T}_\Pi$, and the former can be found using off-the-shelf tools for discovery of automorphisms in explicit graphs, such as BLISS (Junttila and Kaski 2007).

Pochter et al. define the PDG for the SAS$^+$ formalism (Bäckström and Klein 1991). A STRIPS task can be viewed as a SAS$^+$ task over binary-domain state variables with operator preconditions and the goal restricted to variables assigned the value 1. Thus, we present here an adaptation of the definition by Pochter et al. to STRIPS planning tasks.

Definition 1 *Let $\Pi = \langle P, O, I, G, C\rangle$ be a* STRIPS *planning task. The* ***problem description graph*** *(PDG) of Π is the colored digraph $\langle N, E\rangle$ with nodes*
$$N = \bigcup_{p \in P} \{v_p, v_p^{\mathrm{T}}, v_p^{\mathrm{F}}\} \cup \{v_o \mid o \in O\},$$
node colors
$$col(v) = \begin{cases} 1 & \textit{if } v = v_p^{\mathrm{T}}, p \in G \\ 2 + C(o) & \textit{if } v = v_o, o \in O \\ 0 & \textit{otherwise} \end{cases}$$
and edges
$$E = \bigcup_{p \in P} \{\langle v_p, v_p^{\mathrm{T}}\rangle, \langle v_p, v_p^{\mathrm{F}}\rangle\} \cup \bigcup_{o \in O} \left(E_o^{pre} \cup E_o^{add} \cup E_o^{del}\right),$$
where
$$E_o^{pre} = \{\langle v_p^{\mathrm{T}}, v_o\rangle \mid p \in pre(o)\},$$
$$E_o^{add} = \{\langle v_o, v_p^{\mathrm{T}}\rangle \mid p \in add(o)\},$$
$$E_o^{del} = \{\langle v_o, v_p^{\mathrm{F}}\rangle \mid p \in del(o)\}.$$

A *PDG symmetry* is a symmetry of $\mathcal{T}_\Pi$ that is induced by a graph automorphism of the PDG of Π. In the following section, we introduce a more direct definition for symmetries of planning tasks based on the factored task representation. Due to its simplicity, this new definition is easier to reason about than PDG symmetries.

Structural Symmetries

By a *permutation* of a planning task, we mean a permutation of its propositions and operators. Based on this concept, we can directly give a structural notion of symmetry.

Definition 2 *Let $\Pi = \langle P, O, I, G, C\rangle$ be a* STRIPS *planning task. A permutation σ of Π is a* ***structural symmetry*** *if*
- *$\sigma(P) = P$*
- *$\sigma(O) = O$, and for all $o \in O$:*
  - *$pre(\sigma(o)) = \sigma(pre(o))$*
  - *$add(\sigma(o)) = \sigma(add(o))$*
  - *$del(\sigma(o)) = \sigma(del(o))$*
  - *$C(\sigma(o)) = C(o)$*
- *$\sigma(G) = G$*

Intuitively, a structural symmetry preserves (leaves invariant) all aspects of a planning task other than the initial state. By "renaming" propositions and operators, we end up with an identical planning task. The following result establishes that structural symmetries induce transition graph symmetries and shows how they are related to PDG symmetries:

Theorem 2 *Let Π be a planning task. Then:*

1. *If σ is a structural symmetry of Π, then σ (viewed as a function on the states and operators of Π) is a transition graph symmetry of $\mathcal{T}_\Pi$.*
2. *The structural symmetries form a subgroup of $Aut(\mathcal{T}_\Pi)$.*
3. *Every structural symmetry of Π corresponds to a PDG symmetry of Π in the sense that they induce the same transition graph symmetry.*
4. *If each proposition of Π occurs as an operator precondition or in the goal, then every PDG symmetry of Π corresponds to a structural symmetry of Π in the sense that they induce the same transition graph symmetry.*

For space reasons, we refer to a technical report (Shleyfman et al. 2014) for the proof, which is not complicated. Statements 1. and 2. establish that structural symmetries are indeed symmetries and induce an equivalence relation. Because of statements 3. and 4., existing algorithms for detecting PDG symmetries can be used to derive structural symmetries, while proving properties of symmetries based on the much simpler definition of structural symmetry.[1]

In the following, we analyze how planning heuristics from the literature based on delete relaxation, critical paths and landmarks interact with structural symmetries. For a planning task Π and symmetry σ, we say that a given heuristic function h is *invariant under σ* if $h(s) = h(\sigma(s))$ for all states s. We say that a family of heuristic functions is *invariant under structural symmetry* if this invariance property holds for all structural symmetries of all planning tasks.

Delete Relaxation Heuristics

Delete relaxation has played a major role in recent advances of planning systems (e. g., Hoffmann and Nebel 2001; Helmert and Domshlak 2009). In heuristics based on delete relaxation, heuristic values are computed with respect to a modified planning task Π^+, in which all delete effects of actions are omitted (e. g., Hoffmann and Nebel 2001). A plan for the delete relaxation of Π is called a *relaxed plan* for Π.

The *optimal delete relaxation heuristic*, denoted by h^+, assigns to each state s the cost of an optimal relaxed plan for s: $h^+(s, \Pi) := h^*(s, \Pi^+)$. This is an *admissible* estimate (lower bound) for $h^*(s)$. Our first result deals with h^+.

Theorem 3 *The optimal delete relaxation heuristic h^+ is invariant under structural symmetry.*

Proof: Let Π be a planning task, s a state of Π and σ a structural symmetry of Π. Then σ is also a structural symmetry of Π^+: clearly, $del(\sigma(o)) = \sigma(del(o)) = \emptyset$ for all operators o of Π^+, and in all other aspects, Π and Π^+ are identical.

We obtain: $h^+(s, \Pi) \overset{(*)}{=} h^*(s, \Pi^+) \overset{(**)}{=} h^*(\sigma(s), \Pi^+) \overset{(*)}{=} h^+(\sigma(s))$, where (*) holds by definition of h^+ and (**) holds because σ is a structural symmetry of Π^+ and h^* is invariant under structural symmetry (cf. Theorem 1). □

Most planning algorithms based on delete relaxation do not use the h^+ heuristic directly because its computation is NP-equivalent (Bylander 1994). Instead, they use various approximations. We consider three such approximations here: the $h^{\max}$ and h^{add} heuristics by Bonet and Geffner (2001), and the FF heuristic by Hoffmann and Nebel (2001). There are many equivalent ways of defining these heuristics. Our presentation broadly follows Keyder and Geffner (2008), with some differences in details to simplify the proofs and discussion. In this approach, the heuristics are declaratively defined as systems of equations.

For the $h^{\max}$ heuristic for state s of planning task $\Pi = \langle P, O, I, G, C\rangle$, the equations are:

$$propcost(p, s) = 0 \quad \text{if } p \in s \tag{1}$$

$$propcost(p, s) = opcost(supp(p, s), s) \quad \text{if } p \notin s \tag{2}$$

$$supp(p, s) \in \operatorname*{argmin}_{o \in O : p \in add(o)} opcost(o, s) \quad \text{if } p \notin s \tag{3}$$

$$opcost(o, s) = C(o) + setcost(pre(o), s) \tag{4}$$

$$setcost(F, s) = \max_{p \in F} propcost(p, s) \tag{5}$$

$$h^{\max}(s) = setcost(G, s) \tag{6}$$

Here, $propcost(p, s) \in \mathbb{R}^{0+} \cup \{\infty\}$ estimates the cost of reaching proposition $p \in P$ from state s, $setcost(F, s) \in \mathbb{R}^{0+} \cup \{\infty\}$ estimates the cost of reaching the set of propositions $F \subseteq P$ from state s, $opcost(o, s) \in \mathbb{R}^{0+} \cup \{\infty\}$ estimates the cost of reaching a state where o is applicable and then applying it, and $supp(p, s) \in O$ is a *best supporter* of proposition $p \in P$, i. e., an operator which is estimated to offer the cheapest way of achieving p.

If all operator costs are strictly positive, there exists exactly one solution to this set of equations, i. e., exactly one way to define *propcost*, *opcost*, etc. to satisfy the equations, except that there may be multiple minimizers in (3). However, this ambiguity does not affect the heuristic value because all minimizers have the same *opcost*.[2]

We can now present our result for $h^{\max}$.

Theorem 4 *The maximum heuristic $h^{\max}$ is invariant under structural symmetry.*

Proof: Let *propcost*, *opcost*, *setcost* and *supp* be a solution to the equations defining $h^{\max}$. Let σ be a structural symmetry. We define primed versions *propcost*′, *opcost*′, *setcost*′ and *supp*′ as follows:

$$propcost'(p, s) := propcost(\sigma(p), \sigma(s))$$

$$opcost'(o, s) := opcost(\sigma(o), \sigma(s))$$

$$setcost'(F, s) := setcost(\sigma(F), \sigma(s))$$

$$supp'(p, s) := \sigma^{-1}(supp(\sigma(p), \sigma(s)))$$

[1] The requirement in statement 4. that each proposition occurs in a precondition or in the goal is not limiting: propositions violating this requirement can be removed from the planning task without affecting its semantics, and existing commonly used relevance analysis algorithms (Gazen and Knoblock 1997) already do this.

[2] To keep the presentation short, we gloss over some details here: the case of zero-cost actions and the case where the equations minimize or maximize over empty sets are discussed in the technical report (Shleyfman et al. 2014).

We will show that *propcost*′, *opcost*′ etc. also satisfy the equations defining $h^{\max}$. Due to the uniqueness of the solution (except for *supp*), this implies that the primed and unprimed values are identical. Exploiting this for *setcost* in particular, we obtain: $h^{\max}(s) = \mathit{setcost}(G, s) = \mathit{setcost}'(G, s) = \mathit{setcost}(\sigma(G), \sigma(s)) = \mathit{setcost}(G, \sigma(s)) = h^{\max}(\sigma(s))$ where we apply, in sequence: the definition of $h^{\max}$, the equality of *setcost* and *setcost*′, the definition of *setcost*′, the fact $G = \sigma(G)$, and the definition of $h^{\max}$.

It remains to show that *propcost*′, *opcost*′ etc. satisfy Equations (1–5):

(1) If $p \in s$, then $\sigma(p) \in \sigma(s)$, and hence $\mathit{propcost}'(p, s) = \mathit{propcost}(\sigma(p), \sigma(s)) = 0$

(2) If $p \notin s$, then $\sigma(p) \notin \sigma(s)$, and hence:
$\mathit{propcost}'(p, s)$
$= \mathit{propcost}(\sigma(p), \sigma(s))$
$= \mathit{opcost}(\mathit{supp}(\sigma(p), \sigma(s)), \sigma(s))$
$= \mathit{opcost}(\sigma(\sigma^{-1}(\mathit{supp}(\sigma(p), \sigma(s)), \sigma(s))))$
$= \mathit{opcost}(\sigma(\mathit{supp}'(p, s)), \sigma(s))$
$= \mathit{opcost}'(\mathit{supp}'(p, s), s)$

(3) For $p \notin s$, we have $\sigma(p) \notin \sigma(s)$ and obtain:
$\sigma(\mathit{supp}'(p, s))$
$= \sigma(\sigma^{-1}(\mathit{supp}(\sigma(p), \sigma(s))))$
$= \mathit{supp}(\sigma(p), \sigma(s))$
$\in \mathrm{argmin}_{o' \in O : \sigma(p) \in \mathit{add}(o')}\, \mathit{opcost}(o', \sigma(s))$
$= \mathrm{argmin}_{o' \in O : o' = \sigma(o), \sigma(p) \in \mathit{add}(\sigma(o))}\, \mathit{opcost}(\sigma(o), \sigma(s))$
$= \mathrm{argmin}_{o' \in O : o' = \sigma(o), p \in \mathit{add}(o)}\, \mathit{opcost}'(o, s)$
$= \sigma(\mathrm{argmin}_{o \in O : p \in \mathit{add}(o)}\, \mathit{opcost}'(o, s))$ and hence
$\mathit{supp}'(p, s) \in \mathrm{argmin}_{o \in O : p \in \mathit{add}(o)}\, \mathit{opcost}'(o, s)$

(4) $\mathit{opcost}'(o, s)$
$= \mathit{opcost}(\sigma(o), \sigma(s))$
$= C(\sigma(o)) + \mathit{setcost}(\mathit{pre}(\sigma(o)), \sigma(s))$
$= C(o) + \mathit{setcost}(\sigma(\mathit{pre}(o)), \sigma(s))$
$= C(o) + \mathit{setcost}'(\mathit{pre}(o), s)$

(5) $\mathit{setcost}'(F, s)$
$= \mathit{setcost}(\sigma(F), \sigma(s))$
$= \max_{p' \in \sigma(F)} \mathit{propcost}(p', \sigma(s))$
$= \max_{\sigma(p) \in \sigma(F)} \mathit{propcost}(\sigma(p), \sigma(s))$
$= \max_{p \in F} \mathit{propcost}'(p, s)$

□

A corresponding result for h^{add} is now easy to obtain.

Theorem 5 *The additive heuristic h^{add} is invariant under structural symmetry.*

Proof: The definition of h^{add} is identical to $h^{\max}$ except that (5) is replaced by $\mathit{setcost}(F, s) = \sum_{p \in F} \mathit{propcost}(p, s)$. The preceding proof works with the adaptation of replacing the maximum by a sum in the part dealing with (5). □

More generally, a corresponding result holds for all variations of $h^{\max}$ and h^{add} which are obtained by changing the definition of set costs (i. e., using an aggregation function other than maximum or sum). The proof only relies on the fact that the heuristic is well-defined (the equations have a unique solution) and that $\mathit{setcost}(F, s)$ can be defined in terms of the multi-set of fact costs $\{\mathit{propcost}(p, s) \mid p \in F\}$.

Another famous approximation of h^+ is the FF heuristic (Hoffmann and Nebel 2001). Like $h^{\max}$ and h^{add}, it can be defined in terms of best supporters of propositions. Different variants of the FF heuristic exist. The variants most commonly used for cost-based planning are called FF/$h^{\max}$ and FF/h^{add} and compute best supporters in the same way as $h^{\max}$ and h^{add}, respectively (Keyder and Geffner 2008). We focus on the FF/$h^{\max}$ variant in the following, but identical results can be proved for FF/h^{add}. FF/$h^{\max}$ uses Equations (1–5) of the definition of $h^{\max}$ and adds the following ones:

$$\mathit{plan}(p, s) = \emptyset \quad \text{if } p \in s \tag{7}$$

$$\mathit{plan}(p, s) = \{\mathit{supp}(p, s)\} \cup \bigcup_{q \in \mathit{pre}(\mathit{supp}(p, s))} \mathit{plan}(q, s) \quad \text{if } p \notin s \tag{8}$$

$$h^{\text{FF}}(s) = \sum_{o \in \bigcup_{q \in G} \mathit{plan}(q, s)} C(o) \tag{9}$$

The basic idea is to associate with each proposition p a set of operators $\mathit{plan}(p, s)$ which is sufficient to achieve p in Π^+ from state s. Operator sets for sets of propositions are then aggregated by set union, and the overall heuristic value is the total cost of all operators in the operator set associated with the goal.

Unlike $h^{\max}$ and h^{add}, however, the FF heuristic is not well-defined: different ways of choosing between minimizers in the equation for $\mathit{supp}(p, s)$ can lead to different heuristic values. For this reason, if we make no further assumption on the tie-breaking policy used, the FF heuristic is not necessarily invariant under structural symmetry. However, this is not unexpected for a heuristic that is not well-defined. One natural tie-breaking policy is to pick uniformly randomly between multiple minimizers in (3). This turns $\mathit{plan}(p, s)$ and $h^{\text{FF}}(s)$ into well-defined random variables.[3] We can then prove the following result:

Theorem 6 *1. There exist tie-breaking policies for which FF/$h^{\max}$ is not invariant under structural symmetry.*

2. There exist tie-breaking policies for which FF/h^{add} is not invariant under structural symmetry.

3. Let h^{FF} be a randomized variant of the FF heuristic where supporters are selected w.r.t. a heuristic that is invariant under structural symmetry (like $h^{\max}$ or h^{add}) and ties are broken uniformly randomly. This heuristic is invariant under structural symmetry in the sense that for all states s and structural symmetries σ, $h^{\text{FF}}(s)$ and $h^{\text{FF}}(\sigma(s))$ are identically distributed random variables.

Proof sketch: The first two results can be shown by providing an example, for which we refer to the technical report (Shleyfman et al. 2014).

The proof for the third result works on similar principles as the one for Theorem 4. The key step is to show that the random variable $\mathit{plan}'(p, s) := \sigma^{-1}(\mathit{plan}(\sigma(p), \sigma(s)))$ is identically distributed to $\mathit{plan}(p, s)$. Again we refer to the technical report for details. □

[3] Note that $\mathit{plan}(p, s)$ and $\mathit{plan}(q, s)$ for $p \neq q$ are often not independent.

Critical Path Heuristics

The critical path heuristics h^m (Haslum and Geffner 2000) generalize the $h^{\max}$ heuristic. They are parameterized with a natural number $m \geq 1$, and for $m = 1$ we obtain $h^1 = h^{\max}$. For $m > 1$, h^m is no longer bounded by h^+, and for sufficiently large m we have $h^m = h^*$. However, the computational complexity of h^m is exponential in m, and thus in practice m is severely restricted (usually to $m = 2$).

An alternative view of h^m was suggested by Haslum (2009), stating that h^m can be computed as the $h^{\max}$ heuristic of a transformed problem, called Π^m. For convenience, we repeat Haslum's definition (adapted to our notation).

Definition 3 (Haslum, 2009) *Let $\Pi = \langle P, O, I, G, C\rangle$ be a propositional* STRIPS *task and let $m \geq 1$. The* STRIPS *task $\Pi^m = \langle P^m, O^m, I^m, G^m, C^m\rangle$ is defined as follows. For any set $X \subseteq P$, let $S_m(X) = \{c \subseteq X \mid |c| \leq m\}$ denote all sets of at most m elements of X. Then P^m contains a meta-atom π_c for each $c \in S_m(P)$. For each operator $o \in O$ and for each set $f \subseteq P$ with $|f| \leq m{-}1$ such that f is disjoint from $add(o) \cup del(o)$, O^m contains a meta-operator $\alpha_{o,f}$ with:*

$$\begin{aligned} pre(\alpha_{o,f}) &= \{\pi_c \mid c \in S_m(pre(o) \cup f)\} \\ add(\alpha_{o,f}) &= \{\pi_c \mid c \in S_m(add(o) \cup f)\} \\ del(\alpha_{o,f}) &= \emptyset \\ C^m(\alpha_{o,f}) &= C(o) \end{aligned}$$

The initial state is $I^m = \{\pi_c \mid c \in S_m(I)\}$, and the goal is $G^m = \{\pi_c \mid c \in S_m(G)\}$.

Our definition differs from Haslum's in including more add effects in the meta-operators: Haslum additionally requires $c \cap add(o) \neq \emptyset$. By removing this condition, our meta-operators include additional effects that are already preconditions, which clearly does not affect their semantics.

Given a structural symmetry σ of Π, we define a mapping σ^m on the propositions and operators of Π^m as follows. For each meta-atom π_c, we set $\sigma^m(\pi_c) = \pi_{\sigma(c)}$. For each meta-operator $\alpha_{o,f}$, we set $\sigma^m(\alpha_{o,f}) = \alpha_{\sigma(o),\sigma(f)}$. We now show that this mapping is a structural symmetry of Π^m.

Theorem 7 *Let $\Pi = \langle P, O, I, G, C\rangle$ be a* STRIPS *planning task and σ be a structural symmetry of Π. Then σ^m is a structural symmetry of $\Pi^m = \langle P^m, O^m, I^m, G^m, C^m\rangle$.*

Proof: For simplicity, we identify meta-atoms π_c with their proposition sets c in the following. It is easy to verify $\sigma^m(S_m(X)) = S_m(\sigma(X))$ for all $X \subseteq P$. From this we immediately obtain $\sigma^m(P^m) = P^m$ and $\sigma^m(G^m) = G^m$.

We next show $\sigma^m(O^m) = O^m$. Here, we mainly need to verify $\sigma^m(\alpha_{o,f}) \in O^m$ for all $\alpha_{o,f} \in O^m$. Then σ^m clearly defines a bijection on O^m: it is easy to see that it is injective, and hence by a counting argument it must also be bijective.

Consider $\alpha_{o,f} \in O^m$. By definition of O^m, we have $o \in O$ and $f \subseteq P$ with $|f| \leq m-1$ and f is disjoint from $add(o) \cup del(o)$. Then clearly $\sigma(o) \in O$, and $\sigma(f)$ is disjoint from $add(\sigma(o)) \cup del(\sigma(o))$ because σ is a structural symmetry of Π. Hence $\sigma^m(\alpha_{o,f}) = \alpha_{\sigma(o),\sigma(f)} \in O^m$ as desired.

It remains to show that σ^m preserves the structure of meta-operators. For meta-operator $\alpha_{o,f}$, we have

$$\begin{aligned} pre(\sigma^m(\alpha_{o,f})) &= \{\pi_c \mid c \in S_m(pre(\sigma(o)) \cup \sigma(f))\} \\ &= \{\sigma^m(\pi_c) \mid c \in S_m(pre(o) \cup f)\} \\ &= \sigma^m(pre(\alpha_{o,f})) \\ add(\sigma^m(\alpha_{o,f})) &= \{\pi_c \mid c \in S_m(add(\sigma(o)) \cup \sigma(f))\} \\ &= \{\sigma^m(\pi_c) \mid c \in S_m(add(o) \cup f)\} \\ &= \sigma^m(add(\alpha_{o,f})) \\ del(\sigma^m(\alpha_{o,f})) &= \emptyset = \sigma^m(del(\alpha_{o,f})) \\ C(\sigma^m(\alpha_{o,f})) &= C(\alpha_{\sigma(o),\sigma(f)}) \\ &= C(\sigma(o)) = C(o) = C(\alpha_{o,f}) \end{aligned}$$

We can now prove the invariance of h^m. □

Theorem 8 *The heuristic h^m is invariant under structural symmetry.*

Proof: Applying Theorems 4 and 7, we get

$$\begin{aligned} h^m(s, \Pi) &= h^{\max}(S_m(s), \Pi^m) \\ &= h^{\max}(\sigma^m(S_m(s)), \Pi^m) \\ &= h^{\max}(S_m(\sigma(s)), \Pi^m) \\ &= h^m(\sigma(s), \Pi), \end{aligned}$$

where the relationship between h^m in Π and $h^{\max}$ in Π^m is due to Theorem 5 of Haslum (2009). □

Landmark Heuristics

Landmarks were original introduced by Porteous, Sebastia, and Hoffmann (2001) and later revisited by the same authors (Hoffmann, Porteous, and Sebastia 2004). Richter, Helmert, and Westphal (2008) first suggested their use for search heuristics, and since then many heuristics based on landmarks have been suggested (e. g., Karpas and Domshlak 2009; Helmert and Domshlak 2009; Keyder, Richter, and Helmert 2010; Bonet and Helmert 2010).

Definition 4 *Let $\Pi = \langle P, O, I, G, C\rangle$ be a* STRIPS *task. A set of operators $L \subseteq O$ is a **disjunctive action landmark** (**landmark** for short) for state s if every plan π for s includes an operator in L.*

The originally suggested landmark heuristics (Richter, Helmert, and Westphal 2008; Karpas and Domshlak 2009) are *path-dependent*, i. e., their heuristic estimates depend not just on the state s to be evaluated, but also on the path(s) from the initial state to s explored by the search algorithm. Domshlak et al. (2013) already discuss the relationship of symmetries and path-dependent landmark heuristics, so we focus on other aspects of landmark heuristics here, beginning with the following basic result.

Theorem 9 *Let Π be a planning task and σ be a transition graph symmetry of Π. Let s be a state of Π and L be a landmark for s. Then $\sigma(L)$ is a landmark for $\sigma(s)$.*

Proof: Follows immediately from the definition of landmarks and Theorem 1. □

Note that this result applies to general transition graph symmetries, not just to structural symmetries. It is easy to see that an analog of the theorem holds for *action landmarks* (landmarks L with $|L| = 1$) and also for the case of structural symmetries and (disjunctive or non-disjunctive) *fact landmarks* (landmarks L consisting of all achievers of a given set of propositions). The latter result was already shown by Domshlak et al. (2013).

For practical use, however, it is not just important that a given set of operators *is* a landmark. It is also necessary that a planning algorithm can *generate* the landmark in order to exploit it for heuristic information. A *landmark generation method* $\mathcal{L}$ is an algorithm that, given a state s, computes a set of landmarks $\mathcal{L}(s)$ for s. Such algorithms are generally *sound* (produce only landmarks), but not *complete* (do not produce all landmarks). We say that a landmark generation method is *invariant under structural symmetry* if, for all structural symmetries σ, it guarantees $\mathcal{L}(\sigma(s)) = \sigma(\mathcal{L}(s))$.

Current planning algorithms from the literature typically use one of the following landmark generation methods: *ZG* (Zhu and Givan 2003), *RHW* (Richter, Helmert, and Westphal 2008), *KRH* (Keyder, Richter, and Helmert 2010), or some variation of *justification graph landmarks* (e. g., Helmert and Domshlak 2009; Bonet and Helmert 2010; Bonet and Castillo 2011). For space reasons, the algorithms are not described in detail, but our previous discussion is already sufficient to sketch the proof of the following result.

Theorem 10 *The ZG, KRH and complete justification graph landmark generation methods are invariant under structural symmetry.*

Proof sketch: Like h^m, KRH is parameterized by a natural number $m \geq 1$. It generates all *causal fact landmarks* of the Π^m planning task discussed in the section on critical-path heuristics, and the result follows from Theorems 7 and 9. (It is not difficult to see that Theorem 9 remains true when restricting attention to causal fact landmarks under structural symmetries.)

ZG is the special case of KRH with $m = 1$.

Justification graph landmarks are closely related to disjunctive action landmarks of the delete-relaxed task Π^+ (Bonet and Helmert 2010), and hence invariance for the (intractable) method that generates *all* such landmarks follows from Theorem 9 and the proof of Theorem 3. □

We remark that an analogous result does *not* hold for RHW landmarks, which partially depend on the first-order PDDL representation of planning tasks, which is not preserved by our structural symmetries defined on propositional STRIPS tasks.

Besides the landmark generation method, the other important aspect of a landmark-based heuristic is how the information from different landmarks is combined to form a heuristic estimate. Here, it is easy to see that the prevalent methods from the literature are invariant under structural symmetry.

Theorem 11 *Let $\mathcal{L}$ be a landmark generation method that is invariant under structural symmetry, and let h be a heuristic such that $h(s)$ derives a heuristic estimate from $\mathcal{L}(s)$ using one of the following techniques:*

1. *counting landmarks (Richter and Westphal 2010)*
2. *summing the minimal operator costs of each landmark (Richter, Helmert, and Westphal 2008)*
3. *optimal cost partitioning (Karpas and Domshlak 2009)*
4. *uniform cost partitioning with or without special treatment of action landmarks (Karpas and Domshlak 2009)*
5. *hitting sets (Bonet and Helmert 2010)*

Then h is invariant under structural symmetry.

Proof sketch: The result for 1. follows immediately from the previous theorem, from which 2. is also immediate if we consider that $C(\sigma(o)) = C(o)$. Heuristics 3. and 5. are based on solutions to LPs/IPs which are easily seen to be isomorphic for s and $\sigma(s)$. For 4., we refer to the technical report (Shleyfman et al. 2014). □

We close with a final result concerning the *landmark-cut* heuristic $h^{\text{LM-cut}}$ (Helmert and Domshlak 2009). Like the FF heuristic, $h^{\text{LM-cut}}$ is affected by arbitrary tie-breaking, which can lead to symmetric states having different heuristic values. However, when breaking ties uniformly randomly, we can prove the same randomized result as for h^{FF}.

Theorem 12 *There exist tie-breaking policies for which $h^{\text{LM-cut}}$ is not invariant under structural symmetry.*

LM-cut with uniformly random tie-breaking is invariant under structural symmetry in the sense that for all states s and structural symmetries σ, $h^{\text{LM-cut}}(s)$ and $h^{\text{LM-cut}}(\sigma(s))$ are identically distributed random variables.

Proof sketch: The key steps in the proof are the invariance of $h^{\max}$ (Theorem 4), showing that the (randomized) justification graphs computed by LM-cut for s and $\sigma(s)$ are isomorphic, from which it follows that the probability of computing cut L in state s equals the probability of computing cut $\sigma(L)$ in state $\sigma(s)$. With $C(o) = C(\sigma(o))$, we can then show that the heuristic values in each iteration of the LM-cut loop (probabilistically) increase in the same way and the resulting modified planning tasks for the next LM-cut iteration are isomorphic. □

Conclusions

We defined a notion of structural symmetry, which allows directly reasoning about symmetries of a planning task based on its compact representation. We also performed an extensive study of the symmetry properties of existing heuristic functions. Many of the studied heuristics were found to be invariant under structural symmetries, which is encouraging in the sense that it shows that these heuristics do not miss any "obvious" information that could be obtained by reasoning about structural symmetries.

Of the major classes of planning heuristics, our study excluded heuristics based on *abstraction*. These play a major role in cost-optimal classical planning and deserve a separate investigation, which is a subject of future work.

Acknowledgments

This work was supported by the Israel Science Foundation (ISF) grant 1045/12 and by the Swiss National Science Foundation (SNSF) as part of the project "Safe Pruning in Optimal State Space Search" (SPOSSS).

References

Bäckström, C., and Klein, I. 1991. Planning in polynomial time: the SAS-PUBS class. *Computational Intelligence* 7(3):181–197.

Bonet, B., and Castillo, J. 2011. A complete algorithm for generating landmarks. In *Proc. ICAPS 2011*, 315–318.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *AIJ* 129(1):5–33.

Bonet, B., and Helmert, M. 2010. Strengthening landmark heuristics via hitting sets. In *Proc. ECAI 2010*, 329–334.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *AIJ* 69(1–2):165–204.

Culberson, J. C., and Schaeffer, J. 1998. Pattern databases. *Computational Intelligence* 14(3):318–334.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2012. Enhanced symmetry breaking in cost-optimal planning as forward search. In *Proc. ICAPS 2012*.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2013. Symmetry breaking: Satisficing planning and landmark heuristics. In *Proc. ICAPS 2013*, 298–302.

Edelkamp, S. 2001. Planning with pattern databases. In *Proc. ECP 2001*, 84–90.

Emerson, E. A., and Sistla, A. P. 1996. Symmetry and model checking. *Formal Methods in System Design* 9(1–2):105–131.

Felner, A.; Zahavi, U.; Holte, R.; Schaeffer, J.; Sturtevant, N.; and Zhang, Z. 2011. Inconsistent heuristics in theory and practice. *AIJ* 175:1570–1603.

Fox, M., and Long, D. 1999. The detection and exploitation of symmetry in planning problems. In *Proc. IJCAI 1999*, 956–961.

Fox, M., and Long, D. 2002. Extending the exploitation of symmetries in planning. In *Proc. AIPS 2002*, 83–91.

Gazen, B. C., and Knoblock, C. A. 1997. Combining the expressivity of UCPOP with the efficiency of Graphplan. In *Proc. ECP 1997*, 221–233.

Gerevini, A., and Serina, I. 2002. LPG: A planner based on local search for planning graphs with action costs. In *Proc. AIPS 2002*, 13–22.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. In *Proc. AIPS 2000*, 140–149.

Haslum, P. 2009. $h^m(P) = h^1(P^m)$: Alternative characterisations of the generalisation from $h^{\max}$ to h^m. In *Proc. ICAPS 2009*, 354–357.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proc. ICAPS 2009*, 162–169.

Helmert, M.; Haslum, P.; Hoffmann, J.; and Nissim, R. 2014. Merge-and-shrink abstraction: A method for generating lower bounds in factored state spaces. *JACM* 61(3):16:1–63.

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

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR* 14:253–302.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *JAIR* 22:215–278.

Junttila, T., and Kaski, P. 2007. Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX 2007)*, 135–149. SIAM.

Karpas, E., and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *Proc. IJCAI 2009*, 1728–1733.

Katz, M., and Domshlak, C. 2010. Implicit abstraction heuristics. *JAIR* 39:51–126.

Katz, M.; Hoffmann, J.; and Domshlak, C. 2013. Who said we need to relax *all* variables? In *Proc. ICAPS 2013*, 126–134.

Keyder, E., and Geffner, H. 2008. Heuristics for planning with action costs revisited. In *Proc. ECAI 2008*, 588–592.

Keyder, E.; Richter, S.; and Helmert, M. 2010. Sound and complete landmarks for and/or graphs. In *Proc. ECAI 2010*, 335–340.

Korf, R. E., and Felner, A. 2002. Disjoint pattern database heuristics. *AIJ* 134(1–2):9–22.

Pochter, N.; Zohar, A.; and Rosenschein, J. S. 2011. Exploiting problem symmetries in state-based planners. In *Proc. AAAI 2011*, 1004–1009.

Porteous, J.; Sebastia, L.; and Hoffmann, J. 2001. On the extraction, ordering, and usage of landmarks in planning. In *Proc. ECP 2001*, 174–182.

Puget, J.-F. 1993. On the satisfiability of symmetrical constrained satisfaction problems. In *Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems (ISMIS 1993)*, 350–361.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *JAIR* 39:127–177.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In *Proc. AAAI 2008*, 975–982.

Rintanen, J. 2003. Symmetry reduction for SAT representations of transition systems. In *Proc. ICAPS 2003*, 32–40.

Shleyfman, A.; Katz, M.; Helmert, M.; Sievers, S.; and Wehrle, M. 2014. Heuristics and symmetries in classical planning: Additional proofs. Technical Report CS-2014-006, Universität Basel, Departement Mathematik und Informatik.

Zhu, L., and Givan, R. 2003. Landmark extraction via planning graph propagation. In *ICAPS 2003 Doctoral Consortium*, 156–160.