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 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 factorized 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 $\Pi$. The state $I$ is the initial state of $\Pi$. 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 $C : O \rightarrow \mathbb{R}^+_{\geq 0}$ 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 $s[l] := (s \setminus del(o)) \cup add(o)$. The transition graph $T_\Pi = \langle S, E \rangle$ of $\Pi$ is the edge-labeled
digraph over $S$ which contains an edge $\langle s, s' \parallel o \rangle$ from $s$ to $s' \parallel o$ labeled with $o$ whenever $o \in O$ is applicable in state $s$. A sequence of operators $\pi = (o_1, \ldots, o_k)$ is applicable in $s$ if the transition graph contains a path with label sequence $\pi$ starting from $s_{0i}$. 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 operations 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 $T_{\Pi} = (S, E)$ with operators $O$ is a permutation $\sigma$ of $S \cup O$ mapping states to states and operators to operators such that

- $(s, s'; o) \in E$ 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 $\text{Aut}(T_{\Pi})$ of the transition graph. Each subgroup $\Gamma$ 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 $\text{Aut}(G)$ for a graph $G$ is not known to be polynomial-time, but backtracking techniques are surprisingly effective in finding substantial subgroups of $\text{Aut}(G)$.

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

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

**Theorem 1** Let $\Pi$ be a planning task, let $s$ be one of its states, let $\pi$ be a sequence of operators of $\Pi$, and let $\sigma$ be a symmetry of $T_{\Pi}$. Then $\pi$ 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 $\sigma$.

**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 Shleifman 2012; 2013).

As the state transition graph $T_{\Pi}$ of a planning task $\Pi$ is usually too large to be given explicitly, symmetries must be inferred from a compact description. Pochter et al. introduced a method for inferring some symmetries of the planning task from automorphisms of a certain graphical structure, the problem description graph (PDG) of $\Pi$. 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 $\Pi$ induces an automorphism of $T_{\Pi}$, 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 $\Pi = (P, O, I, G, C)$ be a STRIPS planning task. The problem description graph (PDG) of $\Pi$ is the colored digraph $(N, E)$ with nodes

$N = \bigcup_{p \in P} \{v_p, v_p^T, v_p^F\} \cup \{v_o \mid o \in O\}$,

node colors

$\text{col}(v) = \begin{cases} 1 & \text{if } v = v_p^T, p \in G \\ 2 + C(o) & \text{if } v = v_o, o \in O \\ 0 & \text{otherwise} \end{cases}$

and edges

$E = \bigcup_{p \in P} \{\langle v_p, v_p^T \rangle, \langle v_p, v_p^F \rangle\} \cup \bigcup_{o \in O} (E^\text{pre}_o \cup E^\text{add}_o \cup E^\text{del}_o)$,

where

$E^\text{pre}_o = \{\langle v_p^T, v_o \rangle \mid p \in \text{pre}(o)\}$,

$E^\text{add}_o = \{\langle v_o, v_p^F \rangle \mid p \in \text{add}(o)\}$,

$E^\text{del}_o = \{\langle v_o, v_p^F \rangle \mid p \in \text{del}(o)\}$.

A PDG symmetry is a symmetry of $T_{\Pi}$ that is induced by a graph automorphism of the PDG of $\Pi$. 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 $\Pi = (P, O, I, G, C)$ be a STRIPS planning task. A permutation $\sigma$ of $\Pi$ is a structural symmetry if

- $\sigma(P) = P$
- $\sigma(O) = O$, and for all $o \in O$: $\sigma(\text{pre}(o)) = \sigma(\text{pre}(o))$,
- $\sigma(\text{add}(o)) = \sigma(\text{add}(o))$,
- $\sigma(\text{del}(o)) = \sigma(\text{del}(o))$,
- $C(\sigma(o)) = C(o)$
- $\sigma(G) = G$

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:

3372
Theorem 2 Let \( \Pi \) be a planning task. Then:
1. If \( \sigma \) is a structural symmetry of \( \Pi \), then \( \sigma \) (viewed as a function on the states and operators of \( \Pi \)) is a transition graph symmetry of \( T_{\Pi} \).
2. The structural symmetries form a subgroup of \( \text{Aut}(T_{\Pi}) \).
3. Every structural symmetry of \( \Pi \) corresponds to a PDG symmetry of \( \Pi \) in the sense that they induce the same transition graph symmetry.
4. If each proposition of \( \Pi \) occurs as an operator precondition or in the goal, then every PDG symmetry of \( \Pi \) corresponds to a structural symmetry of \( \Pi \) in the sense that they induce the same transition graph symmetry.

For space reasons, we refer to a technical report (Shleymman 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 states are omitted (e.g., Hoffmann and Nebel 2001). A plan \( h \) is the much simpler definition of structural symmetry.

\[ (\ast) \]

Let \( s \) be a structural symmetries of \( \Pi \): clearly, \( \text{del}(\sigma(o)) = \sigma(\text{del}(o)) = \emptyset \) for all operators \( o \) of \( \Pi \), and in all other aspects, \( \Pi \) and \( \Pi^{\ast} \) are identical.

We obtain: \( h^{\ast}(s, \Pi) \subseteq h^{\ast}(s, \Pi^{\ast}) \subseteq h^{\ast}(\sigma(s), \Pi^{\ast}) \subseteq h^{\ast}(\sigma(s)), \) where \( (\ast) \) holds by definition of \( h^{\ast} \) and \( (\ast\ast) \) holds because \( \sigma \) is a structural symmetry of \( \Pi^{\ast} \) and \( h^{\ast} \) is invariant under structural symmetry (cf. Theorem 1).

Most planning algorithms based on delete relaxation do not use the \( h^{\ast} \) heuristic directly because its computation is NP-equivalent (Bylander 1994). Instead, they use various approximations. We consider three such approximations here: the \( h_{\text{max}} \) and \( h_{\text{add}} \) 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 \( h_{\text{max}} \) heuristic for state \( s \) of planning task \( \Pi = \langle P, O, I, G, C \rangle \), the equations are:

\begin{align*}
\text{propcost}(p, s) &= 0 \quad \text{if } p \in s \quad (1) \\
\text{propcost}(p, s) &= \text{opcost}(\text{supp}(p, s), s) \quad \text{if } p \notin s \quad (2) \\
\text{supp}(p, s) &\in \text{argmin } \text{opcost}(o, s) \quad \text{if } p \notin s \quad (3) \\
\text{opcost}(o, s) &= C(o) + \text{setcost}(\text{pre}(o), s) \quad (4) \\
\text{setcost}(F, s) &= \max_{p \in F} \text{propcost}(p, s) \quad (5) \\
\text{h}_{\text{max}}(s) &= \text{setcost}(G, s) \quad (6)
\end{align*}

Here, \( \text{propcost}(p, s) \in \mathbb{R}_{0+}^+ \cup \{\infty\} \) estimates the cost of reaching proposition \( p \in P \) from state \( s \), \( \text{setcost}(F, s) \in \mathbb{R}_{0+}^+ \cup \{\infty\} \) estimates the cost of reaching the set of propositions \( F \subseteq P \) from state \( s \), \( \text{opcost}(o, s) \in \mathbb{R}_{0+}^+ \cup \{\infty\} \) estimates the cost of reaching a state where \( o \) is applicable and then applying it, and \( \text{supp}(p, s) \in O \) 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 \( \text{propcost}, \text{opcost}, \) 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 \( \text{opcost}.\)^2

We can now present our result for \( h_{\text{max}} \).

Theorem 3 The optimal delete relaxation heuristic \( h^{\ast} \) is invariant under structural symmetry.

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

We obtain: \( h^{\ast}(s, \Pi) \subseteq h^{\ast}(s, \Pi^{\ast}) \subseteq h^{\ast}(\sigma(s), \Pi^{\ast}) \subseteq h^{\ast}(\sigma(s)), \) where \( (\ast) \) holds by definition of \( h^{\ast} \) and \( (\ast\ast) \) holds because \( \sigma \) is a structural symmetry of \( \Pi^{\ast} \) and \( h^{\ast} \) is invariant under structural symmetry (cf. Theorem 1).

\[ \Box \]

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 (Shleymman et al. 2014).
We will show that \( \text{propcost}'', \text{opcost}'\) etc. also satisfy the equations defining \( \text{h}^{\max} \). Due to the uniqueness of the solution (except for \( \text{supp} \)), this implies that the primed and unprimed values are identical. Exploiting this for \( \text{setcost} \) in particular, we obtain: \( \text{h}^{\max}(s) = \text{setcost}(G, s) = \text{setcost}'(G, s) = \text{setcost}(\sigma(G), \sigma(s)) = \text{setcost}(G, \sigma(s)) = \text{h}^{\max}(\sigma(s)) \) where we apply, in sequence: the definition of \( \text{h}^{\max} \), the equality of \( \text{setcost} \) and \( \text{setcost}' \), the definition of \( \text{setcost}' \), the fact \( G = \sigma(G) \), and the definition of \( \text{h}^{\max} \).

It remains to show that \( \text{propcost}' \), \( \text{opcost}' \) etc. satisfy Equations (1–5):

1. If \( p \in s \), then \( \sigma(p) \in \sigma(s) \), and hence \( \text{propcost}'(p, s) = \text{propcost}(\sigma(p), \sigma(s)) \).
2. If \( p \notin s \), then \( \sigma(p) \notin \sigma(s) \), and hence:
   \[
   \text{propcost}'(p, s) = \text{propcost}(\sigma(p), \sigma(s)) = \text{opcost}(\text{supp}(\sigma(p), \sigma(s)), \sigma(s)) = \text{opcost}(\sigma^{-1}(\text{supp}(\sigma(p), \sigma(s))), \sigma(s)) = \text{opcost}'(\text{supp}(p, s), s).
   \]
3. For \( p \notin s \), we have \( \sigma(p) \notin \sigma(s) \) and obtain:
   \[
   \text{supp}'(p, s) = \text{supp}(\sigma(p), \sigma(s)) \in \arg\min_{o', \sigma(o') \in \text{add}(o') \setminus \sigma(s)} \text{opcost}(o', \sigma(s)) = \arg\min_{o', \sigma(o') \in \text{add}(o') \setminus \sigma(s)} \text{opcost}(\sigma(o), \sigma(s)) = \arg\min_{o', \sigma(o') \in \text{add}(o') \setminus \sigma(s)} \text{opcost}'(o, s) \quad \text{and hence}
   \]
   \[
   \text{supp}'(p, s) \in \arg\min_{o \in O: \sigma(o) \in \text{add}(o)} \text{opcost}'(o, s).
   \]
4. \( \text{opcost}'(o, s) = \text{opcost}(\sigma(o), \sigma(s)) = C(\sigma(o)) + \text{setcost}(\text{pre}(\sigma(o)), \sigma(s)) = C(o) + \text{setcost}(\text{pre}(\sigma(o)), \sigma(s)) = C(o) + \text{setcost}(\text{pre}(\sigma(o)), s) \).
5. \( \text{setcost}'(F, s) = \text{setcost}(\sigma(F), \sigma(s)) = \max_{p' \in (F \setminus \sigma(F))} \text{propcost}'(p', \sigma(s)) = \max_{p \in F} \text{propcost}'(p, s) \).

A corresponding result for \( \text{h}^{\text{add}} \) is now easy to obtain.

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

**Proof**: The definition of \( \text{h}^{\text{add}} \) is identical to \( \text{h}^{\max} \) except that (5) is replaced by \( \text{setcost}(F, s) = \sum_{p \in F} \text{propcost}(p, s) \). 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 \( \text{h}^{\max} \) and \( \text{h}^{\text{add}} \) 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 \( \text{setcost}(F, s) \) can be defined in terms of the multi-set of fact costs \( \{\text{propcost}(p, s) \mid p \in F\} \).

Another famous approximation of \( \text{h}^+ \) is the FF heuristic (Hoffmann and Nebel 2001). Like \( \text{h}^{\max} \) and \( \text{h}^{\text{add}} \), 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/\( \text{h}^{\max} \) and FF/\( \text{h}^{\text{add}} \) and compute best supporters in the same way as \( \text{h}^{\max} \) and \( \text{h}^{\text{add}} \), respectively (Keyder and Geffner 2008). We focus on the FF/\( \text{h}^{\max} \) variant in the following, but identical results can be proved for FF/\( \text{h}^{\text{add}} \). FF/\( \text{h}^{\max} \) uses Equations (1–5) of the definition of \( \text{h}^{\max} \) and adds the following ones:

\[
\text{plan}(p, s) = \emptyset \quad \text{if} \quad p \in s
\]
\[
\text{plan}(p, s) = \{\text{supp}(p, s)\} \cup \bigcup_{q \in \text{pre}(\text{supp}(p, s))} \text{plan}(q, s) \quad \text{if} \quad p \notin s
\]
\[
\text{h}^{\text{FF}}(s) = \sum_{o \in \bigcup_{q \in \text{plan}(q, s)} \text{opcost}(o)} C(o)
\]

The basic idea is to associate with each proposition \( p \) a set of operators \( \text{plan}(p, s) \) which is sufficient to achieve \( p \) in \( \Pi^+ \) 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 \( \text{h}^{\max} \) and \( \text{h}^{\text{add}} \), however, the FF heuristic is not well-defined: different ways of choosing between minimizers in the equation for \( \text{supp}(p, s) \) 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 \( \text{plan}(p, s) \) and \( \text{h}^{\text{FF}}(s) \) into well-defined random variables.\(^3\) We can then prove the following result:

**Theorem 6** 1. There exist tie-breaking policies for which \( \text{FF}/\text{h}^{\max} \) is not invariant under structural symmetry.
2. There exist tie-breaking policies for which \( \text{FF}/\text{h}^{\text{add}} \) is not invariant under structural symmetry.
3. Let \( \text{h}^{\text{FF}} \) be a randomized variant of the FF heuristic where supporters are selected w.r.t. a heuristic that is invariant under structural symmetry (like \( \text{h}^{\max} \) or \( \text{h}^{\text{add}} \)) and ties are broken uniformly randomly. This heuristic is invariant under structural symmetry in the sense that for all states \( s \) and structural symmetries \( \sigma \), \( \text{h}^{\text{FF}}(s) \) and \( \text{h}^{\text{FF}}(\sigma(s)) \) 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 \( \text{plan}(p, s) := \sigma^{-1}(\text{plan}(\sigma(p), \sigma(s))) \) is identically distributed to \( \text{plan}(p, s) \). Again we refer to the technical report for details.

\(^3\)Note that \( \text{plan}(p, s) \) and \( \text{plan}(q, s) \) for \( p \neq q \) are often not independent.
Critical Path Heuristics
The critical path heuristics $h^m$ (Haslum and Geffner 2000) generalize the $h^\text{max}$ heuristic. They are parameterized with a natural number $m \geq 1$, and for $m = 1$ we obtain $h^1 = h^\text{max}$. 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 $h^\text{max}$ heuristic of a transformed problem, called $\Pi^m$. 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 $S_m(X) = \{c \in X \mid |c| \leq m\}$ denote all sets of at most $m$ elements of $X$. Then $P^m$ contains a meta-atom $\pi_c$ for each $c \in S_m(P)$. For each operator $o \in O$ and for each set $f \subseteq P$ with $|f| < m$ such that $f$ is disjoint from $\text{add}(o) \cup \text{del}(o)$, $O^m$ contains a meta-operator $\alpha_{o,f}$ with:

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

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 $\text{add}(o) \neq \emptyset$. 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 $\sigma$ of $\Pi$, we define a mapping $\sigma^m$ on the propositions and operators of $\Pi^m$ as follows. For each meta-atom $\pi_c$, 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 $\Pi^m$.

Theorem 7 Let $\Pi = \langle P, O, I, G, C \rangle$ be a STRIPS planning task and $\sigma$ be a structural symmetry of $\Pi$. Then $\sigma^m$ 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 $\pi_c$ 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 now 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 $\sigma^m$ 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 $\text{add}(o) \cup \text{del}(o)$. Then clearly $\sigma(o) \in O$, and $\sigma(f)$ is disjoint from $\text{add}(\sigma(o)) \cup \text{del}(\sigma(o))$ because $\sigma$ is a structural symmetry of $\Pi$. Hence $\sigma^m(\alpha_{o,f}) = \alpha_{\sigma(o),\sigma(f)} \in O^m$ as desired.

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

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

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

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

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

Landmark Heuristics
Landmarks were originally 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 $\pi$ 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 $\Pi$ be a planning task and $\sigma$ be a transition graph symmetry of $\Pi$. Let $s$ be a state of $\Pi$ 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 \( \bar{L} \) 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 \( L \) is an algorithm that, given a state \( s \), computes a set of landmarks \( L(s) \) 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 \( \sigma \), it guarantees \( L(\sigma(s)) = \sigma(L(s)) \).

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 \( \Pi^m \) 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 \( \Pi^+ \) (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 \( L \) 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 \( L(s) \) 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/ILPs 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 \( h_{LM-cut} \) (Helmert and Domshlak 2009). Like the FF heuristic, \( h_{LM-cut} \) 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 \( h_{LM-cut} \) is not invariant under structural symmetry.

**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


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


