Heuristics and Symmetries in Classical Planning

Alexander Shleyfman\textsuperscript{1} \quad Michael Katz\textsuperscript{2} \quad Malte Helmert\textsuperscript{3}
Silvan Sievers\textsuperscript{3} \quad Martin Wehrle\textsuperscript{3}

\textsuperscript{1}Technion  
Haifa, Israel

\textsuperscript{2}IBM Haifa Research Lab  
Israel

\textsuperscript{3}University of Basel  
Switzerland
A STRIPS Planning task is 4-tuple \( \langle P, O, I, G \rangle \):

- **P**: finite set of propositions
- **O**: finite set of actions of form \( \langle \text{Pre}, \text{Add}, \text{Del}, c \rangle \) (Preconditions/Add/Delete; subsets of propositions)
  
  \( c \in \mathbb{R}^{0+} \) captures action cost
- **I**: initial state (subset of propositions)
- **G**: goal description (subset of propositions)
Classical Planning as Heuristic Search

Recipe

1. Search algorithm - *BFS* (GBFS or WA*)
2. Heuristic(s) function(s)
3. Secret ingredients:
   - Inference–based state/action pruning
   - Action preferences
   - ...

Introduction
Symmetries
End
Classical Planning as Heuristic Search

Recipe

1. Search algorithm - \textit{BFS} (\textit{GBFS} or \textit{WA}*)
2. Heuristic(s) function(s)
3. Secret ingredients:
   - Inference–based state/action pruning
   - Action preferences
   - ...

Introduction
Symmetries
End
BFS tree step-by-step

$s_0$
BFS tree step-by-step
BFS tree step-by-step

Introduction
Symmetries
End
BFS tree step-by-step
BFS tree step-by-step

\[ s_0 \]

- Diagram showing a breadth-first search tree with root node \( s_0 \) and several levels of children nodes.
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step

Introduction
Symmetries
End
BFS tree step-by-step
BFS tree step-by-step
BFS tree step-by-step

\[ s_0 \]

\[ s_* \]
BFS tree step-by-step
Motivation to use symmetries
Motivation to use symmetries
Motivation to use symmetries
# Exploiting Symmetries for Pruning

**General Recipe**

1. Efficiently generate a **subgroup** of the **automorphism group** of the problem’s transition graph
   - efficiently = empirically efficiently

2. Use that subgroup to prune **some** symmetric states

- Emerson & Sistla (1996) [model checking]
- Rintanen (1993) [planning as SAT]
- Pochter, Zohar, and Rosenschein (2011) [heuristic search]
- Domshlak, Katz, Shleyfman (2012, 2013) [heuristic search]
Exploiting Symmetries for Pruning

**General Recipe**

1. Efficiently generate a subgroup of the automorphism group of the problem’s transition graph
   - efficiently = empirically efficiently
2. Use that subgroup to prune some symmetric states

- Emerson & Sistla (1996) [*model checking*]
- Rintanen (1993) [*planning as SAT*]
- Fox & Long (1999, 2002) [*Graphplan-style*]
- Pochter, Zohar, and Rosenschein (2011) [*heuristic search*]
- Domshlak, Katz, Shleyfman (2012, 2013) [*heuristic search*]
Exploiting Symmetries for Pruning

**General Recipe**

1. Efficiently generate a subgroup of the **automorphism group** of the problem's transition graph
   - efficiently = empirically efficiently
2. Use that subgroup to prune some symmetric states

- Emerson & Sistla (1996) [*model checking*]
- Rintanen (1993) [*planning as SAT*]
- Fox & Long (1999, 2002) [*Graphplan-style*]
- Pochter, Zohar, and Rosenschein (2011) [*heuristic search*]
- Domshlak, Katz, Shleyfman (2012, 2013) [*heuristic search*]
Symmetry Groups

STRIPS
Symmetry Groups

STRIPS

$Aut(\mathcal{T}_\Pi)$
Symmetry Groups

PDG

STRIPS

$\text{Aut}(\mathcal{T}_\Pi)$
Symmetry Groups

Strips

$PDG$

$Aut(PDG)$

$Aut(T_{II})$
Symmetry Groups

PDG

STRIPS

Aut(PDG)
Symmetry Groups

Introduction
Symmetries
Symmetric Heuristics
End
Definition

Let $\langle P, O, I, G \rangle$ be a STRIPS planning task. A permutation $\sigma$ is a structural symmetry if

$\sigma(P) = P$

$\sigma(O) = O$, and for all $o \in O$:

$Pre(\sigma(o)) = \sigma(Pre(o))$

$Add(\sigma(o)) = \sigma(Add(o))$

$Del(\sigma(o)) = \sigma(Del(o))$

$C'(\sigma(o)) = C(o)$

$\sigma(G') = G$
Born equal?
Why do we want to know?
Why do we want to know?

$A \xrightarrow{\text{symmetry pruning}} A'$

Symmetries
Symmetric Heuristics
End
Why do we want to know?

Why do we want to know?
Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric
<table>
<thead>
<tr>
<th>Non-symmetric</th>
<th>Symmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h^+$ Hoffmann &amp; Nebel</td>
</tr>
</tbody>
</table>
Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric

\( h^+ \)  Hoffmann & Nebel

\( h_{\text{max}} \)  Bonet & Geffner
Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric

\( h^+ \) Hoffmann & Nebel
\( h_{\text{max}} \) Bonet & Geffner
\( h_{\text{add}} \) Bonet & Geffner
### Heuristics Invariance Under Structural Symmetries

<table>
<thead>
<tr>
<th>Non-symmetric</th>
<th>Symmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{FF}$ Hoffmann &amp; Nebel</td>
<td>$h^+$ Hoffmann &amp; Nebel</td>
</tr>
<tr>
<td></td>
<td>$h_{max}$ Bonet &amp; Geffner</td>
</tr>
<tr>
<td></td>
<td>$h_{add}$ Bonet &amp; Geffner</td>
</tr>
<tr>
<td></td>
<td>$E h_{FF}$ Hoffmann &amp; Nebel</td>
</tr>
</tbody>
</table>
Heuristics Invariance Under Structural Symmetries

Non-symmetric

- $h_{FF}$ Hoffmann & Nebel
- $h_{FF}/h_{add}$ Keyder & Geffner

Symmetric

- $h^+$ Hoffmann & Nebel
- $h_{max}$ Bonet & Geffner
- $h_{add}$ Bonet & Geffner
- $\mathbb{E}h_{FF}$ Hoffmann & Nebel
Heuristics Invariance Under Structural Symmetries

### Non-symmetric

- $h_{\text{FF}}$ Hoffmann & Nebel
- $h_{\text{FF}}/h_{\text{add}}$ Keyder & Geffner
- $h_{\text{FF}}/h_{\text{max}}$ Keyder & Geffner

### Symmetric

- $h^+$ Hoffmann & Nebel
- $h_{\text{max}}$ Bonet & Geffner
- $h_{\text{add}}$ Bonet & Geffner
- $\mathbb{E}h_{\text{FF}}$ Hoffmann & Nebel
Heuristics Invariance Under Structural Symmetries

Non-symmetric

- $h_{FF}$ Hoffmann & Nebel
- $h_{FF}/h_{add}$ Keyder & Geffner
- $h_{FF}/h_{max}$ Keyder & Geffner

Symmetric

- $h^+$ Hoffmann & Nebel
- $h_{max}$ Bonet & Geffner
- $h_{add}$ Bonet & Geffner
- $E h_{FF}$ Hoffmann & Nebel
- $h^m$ Haslum & Geffner
Heuristics Invariance Under Structural Symmetries

<table>
<thead>
<tr>
<th>Non-symmetric</th>
<th>Symmetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{FF}$ Hoffmann &amp; Nebel</td>
<td>$h^+$ Hoffmann &amp; Nebel</td>
</tr>
<tr>
<td>$h_{FF}/h_{add}$ Keyder &amp; Geffner</td>
<td>$h_{max}$ Bonet &amp; Geffner</td>
</tr>
<tr>
<td>$h_{FF}/h_{max}$ Keyder &amp; Geffner</td>
<td>$h_{add}$ Bonet &amp; Geffner</td>
</tr>
<tr>
<td>$h^L M$-cut Helmert &amp; Domshlak</td>
<td>$m$ Haslum &amp; Geffner</td>
</tr>
<tr>
<td></td>
<td>$h^L M$-cut Helmert &amp; Domshlak</td>
</tr>
</tbody>
</table>
Landmarks and generation procedures

Non-symmetric

Symmetric

- Given that generation method is invariant under structural symmetry the heuristics below are symmetric
  - counting landmarks (Richter, Helmert, & Westphal)
  - optimal/uniform cost partitioning (Karpas & Domshlak)
  - hitting sets (Bonet & Helmert)
Landmarks and generation procedures

<table>
<thead>
<tr>
<th>Non-symmetric</th>
<th>Symmetric</th>
</tr>
</thead>
</table>
| ZG Zhu & Givan 2003 | }
Landmarks and generation procedures

**Non-symmetric**

**Symmetric**

*ZG* Zhu & Givan 2003

*KRH* Keyder, Richter, & Helmert 2010
<table>
<thead>
<tr>
<th>Landmarks and generation procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-symmetric</strong></td>
</tr>
<tr>
<td><strong>Symmetric</strong></td>
</tr>
<tr>
<td><em>ZG</em> Zhu &amp; Givan 2003</td>
</tr>
<tr>
<td><em>KRH</em> Keyder, Richter, &amp; Helmert 2010</td>
</tr>
</tbody>
</table>

*RHW* Richter, Helmert, & Westphal 2008
Landmarks and generation procedures

Non-symmetric

Symmetric

_G_ Zhu & Givan 2003

_KRH_ Keyder, Richter, & Helmert 2010

_RHW_ Richter, Helmert, & Westphal 2008

Given that generation method is invariant under structural symmetry the heuristics below are symmetric

- counting landmarks (Richter, Helmert, & Westphal)
- optimal/uniform cost partitioning (Karpas & Domshlak)
- hitting sets (Bonet & Helmert)
Lobster is taken from: http://www.biology.ualberta.ca/palmer.hp/asymp/axes/split lobster.GIF