A Comparison of Cost Partitioning Algorithms for Optimal Classical Planning

Jendrik Seipp   Thomas Keller   Malte Helmert

University of Basel

June 21, 2017
• optimal classical planning
• A* search + admissible heuristic
• abstraction heuristics
• optimal classical planning
• A∗ search + admissible heuristic
• abstraction heuristics
• single heuristic unable to capture enough information
Problem

- single heuristic unable to capture enough information
  → use multiple heuristics
Problem

- single heuristic unable to capture enough information
  → use multiple heuristics
- how to combine multiple heuristics admissibly?
Multiple Heuristics

\[ h_1(s_1) = 5 \]
\[ h_2(s_1) = 5 \]

- Maximizing only selects best heuristic

\[ h(s_1) = 5 \]
Multiple Heuristics

$h_1(s_1) = 5$

$h_2(s_1) = 5$
Multiple Heuristics

- $h_1(s_1) = 5$
- $h_2(s_1) = 5$

- maximizing only selects best heuristic $\rightarrow h(s_1) = 5$
Multiple Heuristics: Cost Partitioning

Cost Partitioning

- split operator costs among heuristics
- total costs must not exceed original costs

→ combines heuristics

→ allows summing heuristic values admissibly
Cost Partitioning Algorithms

Optimal Cost Partitioning

- cost partitioning with highest heuristic value for a given state among all cost partitionings
- computable in polynomial time for abstractions
- too expensive in practice

$h(s_1) = ?$
Cost Partitioning Algorithms

Optimal Cost Partitioning

- cost partitioning with highest heuristic value for a given state among all cost partitionings
- computable in polynomial time for abstractions
- too expensive in practice

$h(s_1) = 5 + 3 = 8$
Cost Partitioning Algorithms

Post-hoc Optimization

- compute best factor $0 \leq w \leq 1$ for each heuristic
- for each operator: sum of relevant heuristic factors $\leq 1$
  e.g., $w_1 + w_2 \leq 1$, $w_2 \leq 1$
- use costs $w \cdot \text{cost}(o)$ if $o$ is relevant for $h$, otherwise 0

$h(s_1) = ?$
Cost Partitioning Algorithms

*Post-hoc Optimization*

- compute best factor $0 \leq w \leq 1$ for each heuristic
- for each operator: sum of relevant heuristic factors $\leq 1$
  - e.g., $w_1 + w_2 \leq 1$, $w_2 \leq 1$
- use costs $w \cdot \text{cost}(o)$ if $o$ is relevant for $h$, otherwise 0

\[ w_1 = 0.25, \quad w_2 = 0.75 \rightarrow h(s_1) = 1.25 + 3.75 = 5 \]
Cost Partitioning Algorithms

Greedy Zero-one Cost Partitioning

- order heuristics
- use full costs for the first relevant heuristic
Cost Partitioning Algorithms

Greedy Zero-one Cost Partitioning

- order heuristics
- use full costs for the first relevant heuristic

\[ h(s_1) = 5 + 0 = 5 \]
Saturated Cost Partitioning

- order heuristics
- for each heuristic $h$:
  - use minimum costs preserving all heuristic estimates for $h$
  - use remaining costs for subsequent heuristics

$h(s_1) = ?$
**Saturated Cost Partitioning**

- order heuristics
- for each heuristic $h$:
  - use minimum costs preserving all heuristic estimates for $h$
  - use remaining costs for subsequent heuristics

\[ h(s_1) = 5 + 3 = 8 \]
Cost Partitioning Algorithms

Uniform Cost Partitioning
- distribute costs uniformly among relevant heuristics

$h(s_1) = \ ???$
Cost Partitioning Algorithms

\[ h(s_1) = 3 + 3 = 6 \]

Uniform Cost Partitioning

- distribute costs uniformly among relevant heuristics
Cost Partitioning Algorithms

\[ h(s_1) = ? \]

Opportunistic Uniform Cost Partitioning (New)

- order heuristics
- for each heuristic \( h \):
  - distribute costs uniformly among \( h \) and other relevant remaining heuristics
  - use saturated costs for \( h \)
  - use remaining costs for subsequent heuristics
Cost Partitioning Algorithms

Cost Partitioning Algorithms

\[ h(s_1) = 3 + 4 = 7 \]

Opportunistic Uniform Cost Partitioning (New)

- order heuristics
- for each heuristic \( h \):
  - distribute costs uniformly among \( h \) and other relevant remaining heuristics
  - use saturated costs for \( h \)
  - use remaining costs for subsequent heuristics
Cost Partitioning Algorithms

Canonical Heuristic

- compute independent heuristic subsets
- compute maximum over sums

$h(s_1) = ?$
Cost Partitioning Algorithms

Canonical Heuristic

- compute independent heuristic subsets
- compute maximum over sums

$h(s_1) = \max(5, 5) = 5$
Theoretical Comparison

\[ h^{SCP} \nless\less h^{OUCP} \]

\[ h^{PHO} \nless\less h^{UCP} \]

\[ h^{GZOC} \nless\less h^{CAN} \]

Pommerening et al. 2013

\[ \forall \text{ for } \geq 1 \text{ order} \]
Theoretical Comparison

$\mathcal{h}^{SCP}$ $\mathcal{h}^{GCOP}$

$\mathcal{h}^{PHO}$ $\mathcal{h}^{CAN}$

$\mathcal{h}^{OUCP}$ $\mathcal{h}^{UCP}$

Pommerening et al. 2013

for $\geq 1$ order
Theoretical Comparison

Pommerening et al. 2013

for $\geq 1$ order

Comparison of Cost Partitioning Algorithms
Empirical Comparison

Heuristics:
- hill climbing pattern databases
- systematic pattern databases
- Cartesian abstractions
- landmark heuristics
Empirical Comparison

Heuristics:

- hill climbing pattern databases
- systematic pattern databases
- Cartesian abstractions
- landmark heuristics

Orders:

- for order-dependent algorithms: single order and diverse orders
## Empirical Comparison: Systematic PDBs

<table>
<thead>
<tr>
<th></th>
<th>UCP</th>
<th>OUCP</th>
<th>OUCP</th>
<th>GZOCP</th>
<th>GZOCP</th>
<th>SCP</th>
<th>SCP</th>
<th>CAN</th>
<th>PHO</th>
<th>OCP</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_{\text{UCP}}$</td>
<td>– 0 3 15 3 4 0</td>
<td>11 10 30</td>
<td>709.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{OUCP}}$ <em>single</em></td>
<td>14 9 22 8 6 0</td>
<td>14 13 31</td>
<td>744.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{OUCP}}$ <em>diverse</em></td>
<td>13 8 22 7 6 0</td>
<td>14 14 31</td>
<td>734.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{GZOCP}}$ <em>single</em></td>
<td>3 1 4 – 3 0 0</td>
<td>9 11 29</td>
<td>694.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{GZOCP}}$ <em>diverse</em></td>
<td>15 12 14 20 – 9 0</td>
<td>13 13 30</td>
<td>749.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{SCP}}$ <em>single</em></td>
<td>20 19 17 23 16 – 0</td>
<td>18 21 32</td>
<td>775.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{SCP}}$ <em>diverse</em></td>
<td>27 26 24 28 22 22 – 23 26 33</td>
<td>854.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{CAN}}$</td>
<td>8 7 7 17 5 8 2 – 13 28</td>
<td>656.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{PHO}}$</td>
<td>9 7 7 15 7 6 3 10 – 31</td>
<td>737.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_{\text{OCP}}$</td>
<td>4 4 4 4 4 4 3 5 3 –</td>
<td>471.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Empirical Comparison: Systematic PDBs

Expansions (excluding last \( f \) layer)

\[ h_{\text{SCP}}^{\text{diverse}} \]

\[ h^{\text{PHO}} \]

unsolved

unsolved
Discussion of Results

In each setting:

- **reuse** unused costs
- **assign costs** greedily
- **use** multiple orders

→ saturated cost partitioning
Comparison to State of the Art (Using $h^2$ Mutexes)

<table>
<thead>
<tr>
<th></th>
<th>HC+$h^\text{SCP}_{\text{diverse}}$</th>
<th>Sys2+$h^\text{SCP}_{\text{diverse}}$</th>
<th>Cart.$+h^\text{SCP}_{\text{diverse}}$</th>
<th>LM+$h^\text{SCP}_{\text{single}}$</th>
<th>SymBA$_2^*$</th>
<th>coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>HC+$h^\text{SCP}_{\text{diverse}}$</td>
<td>– 7 9 19 17</td>
<td>845.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sys2+$h^\text{SCP}_{\text{diverse}}$</td>
<td>10 – 11 18 16</td>
<td>878.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cart.$+h^\text{SCP}_{\text{diverse}}$</td>
<td>19 14 – 24 17</td>
<td>1017.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM+$h^\text{SCP}_{\text{single}}$</td>
<td>8 9 4 – 9</td>
<td>934.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SymBA$_2^*$</td>
<td>20 18 16 23 –</td>
<td>1008.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Better Orders for Saturated Cost Partitioning in Optimal Classical Planning

- combination of three types of abstraction heuristics
- better method for finding heuristic orders
- significantly higher coverage
Conclusion

- new dominance relations
- new cost partitioning algorithm
- saturated cost partitioning preferable in all tested settings