Saturated Post-hoc Optimization for Classical Planning

Jendrik Seipp, Thomas Keller, Malte Helmert

February, 2021

Linköping University (1) and University of Basel (2)
Optimal Classical Planning
Abstraction Heuristics
Abstraction Heuristics
how to combine multiple heuristics?
how to combine multiple heuristics?
Multiple Heuristics

how to combine multiple heuristics?

\[ h_1(s_2) = 5 \]

\[ h_2(s_2) = 4 \]
Multiple Heuristics

how to combine multiple heuristics?

\[ h_1(s_2) = 5 \]
\[ h_2(s_2) = 4 \]

maximize over estimates:

• \( h(s_2) = 5 \)
how to combine multiple heuristics?

- $h_1(s_2) = 5$
- $h_2(s_2) = 4$

maximize over estimates:

- $h(s_2) = 5$
- only selects best heuristic
- does not combine heuristics
Cost Partitioning

- split operator costs among heuristics
- sum of costs must not exceed original cost
Cost Partitioning

- split operator costs among heuristics
- sum of costs must not exceed original cost
Cost Partitioning

- split operator costs among heuristics
- sum of costs must not exceed original cost

\[ h(s_2) = 3 + 3 = 6 \]
Saturated Cost Partitioning
Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

- order heuristics, then for each heuristic $h$:
  - use minimum costs preserving all estimates of $h$
  - use remaining costs for subsequent heuristics
Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

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

![Diagram of Saturated Cost Partitioning Algorithm]
Saturated Cost Partitioning Algorithm

- order heuristics, then for each heuristic $h$:
  - use minimum costs preserving all estimates of $h$
  - use remaining costs for subsequent heuristics
Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

• order heuristics, then for each heuristic $h$:
  • use minimum costs preserving all estimates of $h$
  • use remaining costs for subsequent heuristics
Saturated Cost Partitioning

Saturated Cost Partitioning Algorithm

• order heuristics, then for each heuristic $h$:
  • use minimum costs preserving all estimates of $h$
  • use remaining costs for subsequent heuristics
Saturated Cost Partitioning

**Saturated Cost Partitioning Algorithm**

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

$$h(s_2) = 5 + 3 = 8$$
Post-hoc Optimization
Post-hoc Optimization

\[
\begin{align*}
\minimize \quad & A + B + C + D \\
\text{subject to} \quad & A, B, D \text{ active} \\
& h_1(s_2) = 5 \rightarrow A + B + D \geq 5 \\
& h_2(s_2) = 4 \rightarrow A + B + C \geq 4 \\
& A \geq 0, B \geq 0, C \geq 0, D \geq 0
\end{align*}
\]
Post-hoc Optimization

\[ \text{minimize } 4A + 4B + 1C + 1D \]

\text{such that}

- \( A, B, D \) active
- \( h_1(s_2) = 5 \)

\( h_2(s_2) = 4 \to 4A + 4B + 1C \geq 4 \)

\( h_2(s_2) = 5 \)
Post-hoc Optimization

\[
\begin{align*}
\text{minimize} \quad & 4A + 4B + 1C + 1D \\
\text{such that} \\
& A, B, D \text{ active} \\
& h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5
\end{align*}
\]

- $A, B, D$ active
- $h_1(s_2) = 5$ \quad $\rightarrow$ \quad $4A + 4B + 1D \geq 5$
Post-hoc Optimization

\[
\begin{align*}
\text{minimize} & \quad 4A + 4B + 1C + 1D \\
\text{subject to} & \quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5 \\
& \quad h_2(s_2) = 4
\end{align*}
\]

- \( A, B, D \) active \quad \Rightarrow \quad h_1(s_2) = 5 \quad \rightarrow \quad 4A + 4B + 1D \geq 5
- \( A, B, C \) active \quad \Rightarrow \quad h_2(s_2) = 4
Post-hoc Optimization

\[ \text{minimize } 4A + 4B + 1C + 1D \]

such that

- \( A, B, D \) active \( h_1(s_2) = 5 \) \( \Rightarrow 4A + 4B + 1D \geq 5 \)
- \( A, B, C \) active \( h_2(s_2) = 4 \) \( \Rightarrow 4A + 4B + 1C \geq 4 \)
Post-hoc Optimization

\[ \begin{align*}
A, B, D \text{ active} & \quad \Rightarrow \quad h_1(s_2) = 5 \quad \Rightarrow \quad 4A + 4B + 1D \geq 5 \\
A, B, C \text{ active} & \quad \Rightarrow \quad h_2(s_2) = 4 \quad \Rightarrow \quad 4A + 4B + 1C \geq 4 \\
A \geq 0, B \geq 0, C \geq 0, D \geq 0
\end{align*} \]
Post-hoc Optimization

\[
\text{minimize } 4A + 4B + 1C + 1D \text{ such that }
\]

- $A, B, D$ active \( h_1(s_2) = 5 \) \( \implies 4A + 4B + 1D \geq 5 \)
- $A, B, C$ active \( h_2(s_2) = 4 \) \( \implies 4A + 4B + 1C \geq 4 \)
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$
Post-hoc Optimization

minimize $4A + 4B + 1C + 1D$ such that

- $A$, $B$, $D$ active $h_1(s_2) = 5 \rightarrow 4A + 4B + 1D \geq 5$
- $A$, $B$, $C$ active $h_2(s_2) = 4 \rightarrow 4A + 4B + 1C \geq 4$
- $A \geq 0$, $B \geq 0$, $C \geq 0$, $D \geq 0$

$h(s_2) = 5$
Saturated Post-hoc Optimization
Saturated Post-hoc Optimization

minimize $4A + 4B + 1C + 1D$ such that

- $4A + 4B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$
**Saturated Post-hoc Optimization**

\[ \text{minimize } 4A + 4B + 1C + 1D \text{ such that} \]

- \(4A + 4B + 1D \geq 5\)
- \(4A + 4B + 1C \geq 4\)
- \(A \geq 0, B \geq 0, C \geq 0, D \geq 0\)
minimize $4A + 4B + 1C + 1D$ such that

- $4A + 1B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$
minimize $4A + 4B + 1C + 1D$ such that

- $4A + 1B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$
minimize $4A + 4B + 1C + 1D$ such that

- $4A + 1B + 1D \geq 5$
- $4A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$
Saturated Post-hoc Optimization

\[
\text{minimize } 4A + 4B + 1C + 1D \text{ such that }
\]

- \[4A + 1B + 1D \geq 5\]
- \[1A + 4B + 1C \geq 4\]
- \[A \geq 0, B \geq 0, C \geq 0, D \geq 0\]
minimize $4A + 4B + 1C + 1D$ such that

- $4A + 1B + 1D \geq 5$
- $1A + 4B + 1C \geq 4$
- $A \geq 0, B \geq 0, C \geq 0, D \geq 0$

$h(s_2) = 7.2$
Properties

- admissible
- dominates post-hoc optimization
Relation to Other Cost Partitioning Algorithms
Cost Partitioning Algorithms

Uniform Cost Partitioning

distribute costs evenly among relevant heuristics
Cost Partitioning Algorithms

GZOCP

Greedy Zero-one Cost Partitioning

order heuristics and give full cost to first relevant heuristic
Cost Partitioning Algorithms

- GZOCPP
- PhO
- UCP
- Post-hoc Optimization
Cost Partitioning Algorithms

Canonical Heuristic
maximum over sums of independent heuristic subsets

GZOCP
PhO
CAN
UCP
Cost Partitioning Algorithms

GZOCP

PhO → CAN

UCP

Pommerening et al. 2013
Cost Partitioning Algorithms

GZOCPP
PhO    
CAN
UCP

Seipp et al. 2017
Cost Partitioning Algorithms

SCP

GZOCP

PhO

CAN

UCP
Cost Partitioning Algorithms

SCP ≻≻≻ GZOCP

PhO ≻≻≻ CAN

UCP

Seipp et al. 2017
Cost Partitioning Algorithms

SCP ≻≻≻ GZOCP

PhO ≻≻≻ CAN

OUCP

UCP

Seipp et al. 2017
Cost Partitioning Algorithms

SCP ➚ GZOCPP

PhO ➚ CAN

OUCP ➚ UCP

Seipp et al. 2017
Cost Partitioning Algorithms

SCP $\triangleright\triangleright\triangleright$ GZOCPP

$\triangleright\triangleright\triangleright$

SPhO $\triangleright$ PhO $\triangleright$ CAN

OUCP $\triangleright\triangleright\triangleright$ UCP
Cost Partitioning Algorithms

SCP  \(\gg\gg\gg\)  GZOCP

SPhO  \(\gg\gg\gg\)  PhO  \(\gg\gg\gg\)  CAN

OUCP  \(\gg\gg\gg\)  UCP
$h^{SCP}_{\langle h_1, h_2 \rangle}(s_2) = 8$

$h^{SCP}_{\langle h_2, h_1 \rangle}(s_2) = 7$

$h^{SPhO}(s_2) = 7.2$
Experiments
Experiments: Setup

- saturated PhO vs. PhO
- compute for each state
- hill-climbing PDBs, systematic PDBs, Cartesian Abstractions
- 30 minutes, 3.5 GiB
## Experiments: Coverage

<table>
<thead>
<tr>
<th></th>
<th>Hill Climbing</th>
<th>Systematic</th>
<th>Cartesian</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domains ↑ (48)</td>
<td>6</td>
<td>16</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>Domains ↓ (48)</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Tasks (1827)</td>
<td>823 (+10)</td>
<td>759 (+51)</td>
<td>657 (+169)</td>
<td>806 (+169)</td>
</tr>
</tbody>
</table>
Experiments: Expansions for Combined Abstractions

failed

failed

saturated PhO

PhO

$10^0$ $10^1$ $10^2$ $10^3$ $10^4$ $10^5$ $10^6$

$10^0$ $10^1$ $10^2$ $10^3$ $10^4$ $10^5$ $10^6$
Conclusions

Saturated Post-hoc Optimization

- saturates costs
- dominates original
- admissible
- much stronger heuristics