Getting the Most Out of Pattern Databases for Classical Planning

Florian Pommerening  Gabriele Röger  Malte Helmert

University of Basel, Switzerland

IJCAI 2013
Getting the Most Out of Pattern Databases for Classical Planning
Getting the Most Out of Pattern Databases for Classical Planning
### Structure of this talk

- Classical planning
- PDBs
- iPDB procedure
- Post-hoc optimization heuristic
- Experimental results

---

**Getting the Most Out of Pattern Databases for Classical Planning**
Getting the Most Out of Pattern Databases for Classical Planning
Planning tasks

Planning task

- Variables
  - Variable assignments are states

- Operators
  - Allow to manipulate states
  - Transitions in implicitly defined transition system

- Initial state

- Goal description
  - Find (shortest) path
Solving planning tasks

Common approach

- Informed search algorithm + heuristic

Optimal planning

- A* + admissible heuristic

One type of admissible heuristics

- Pattern database (PDB) heuristics
Pattern database heuristics by example

Pattern database
- Projection to subset of variables
- Abstract distance as heuristic value

Diagram showing the pattern database with nodes labeled from 000 to 311 and edges connecting them, indicating the heuristic values.
Pattern database heuristics by example

Pattern database
- Projection to subset of variables
- Abstract distance as heuristic value
Running example

**Example task**

- Three variables \{A, B, C\}
- Each operator affects only one variable
- Pattern databases

\[
\begin{align*}
h^{\{A\}}(s) &= h^{\{B\}}(s) = h^{\{C\}}(s) = 1 \\
h^{\{A,B\}}(s) &= h^{\{A,C\}}(s) = h^{\{B,C\}}(s) = 6
\end{align*}
\]

What is the best heuristic value we can get from this information?
Using multiple PDBs

**Getting Much Out of Pattern Databases for Classical Planning**

Key idea: Use multiple PDBs

Two aspects

1. Pattern selection
2. Heuristic combination
Getting Much Out of Pattern Databases for Classical Planning

Key idea: Use multiple PDBs

iPDB procedure [Haslum et al.]

1. Pattern selection → hill-climbing search
2. Heuristic combination → canonical heuristic
Additivity of a pattern collection $\mathcal{C}$

- No operator affects variables in two patterns
- Sum of heuristic values is admissible

Canonical heuristic

- Sum where possible, maximize where necessary
- $\text{MAS}(\mathcal{C})$: set of maximal additive subsets of $\mathcal{C}$

Definition (Canonical heuristic)

$$h^\mathcal{C}(s) = \max_{\mathcal{A}\in\text{MAS}(\mathcal{C})} \sum_{P\in\mathcal{A}} h^P(s).$$
Canonical heuristic (example task)

Example task

Each operator affects only one variable

⇒ Disjoint patterns are additive

• For example $h^{\{A\}}(s) + h^{\{B,C\}}(s) = 1 + 6$

$$h^C(s) = 7$$
Post-hoc optimization heuristic: idea

**Getting the Most Out of Pattern Databases for Classical Planning**

→ Can we do better than the canonical heuristic?
Post-hoc optimization heuristic: idea

Example task

\[ h\{A,B\} = 6 \]

\[ \Rightarrow \text{Any solution spends at least cost 6 on operators modifying } A \text{ or } B. \]
Example task

\[ h\{A,B\} = 6 \]

⇒ Any solution spends at least cost 6 on operators modifying A or B.

\[ 6 = h\{A,B\} \leq \text{type-}A + \text{type-}B \]
Post-hoc optimization heuristic: idea

Example task

\[ h\{A,B\} = 6 \]

⇒ Any solution spends at least cost 6 on operators modifying A or B.

\[
6 = h\{A,B\} \leq \text{type-}A + \text{type-}B \\
6 = h\{A,C\} \leq \text{type-}A + \text{type-}C \\
6 = h\{B,C\} \leq \text{type-}B + \text{type-}C
\]
Post-hoc optimization heuristic: idea

Example task

\[ h\{A,B\} = 6 \]

\[ \Rightarrow \text{Any solution spends at least cost } 6 \text{ on operators modifying } A \text{ or } B. \]

\[
\begin{align*}
6 &= h\{A,B\} \\ 
6 &= h\{A,C\} \\ 
6 &= h\{B,C\} \\
18 &\leq 2\text{type-}A + 2\text{type-}B + 2\text{type-}C
\end{align*}
\]
Post-hoc optimization heuristic: idea

**Example task**

\[ h\{A,B\} = 6 \]

⇒ Any solution spends at least cost 6 on operators modifying \( A \) or \( B \).

\[
\begin{align*}
6 &= h\{A,B\} \\
6 &= h\{A,C\} \\
6 &= h\{B,C\}
\end{align*}
\]

\[
\begin{align*}
6 &\leq \text{type-}A + \text{type-}B \\
6 &\leq \text{type-}A + \text{type-}B + \text{type-}C \\
6 &\leq 2\text{type-}A + 2\text{type-}B + 2\text{type-}C
\end{align*}
\]

⇒ at least cost 9 in any plan
Post-hoc optimization heuristic: idea

Example task

\[ h\{A,B\} = 6 \]

⇒ Any solution spends at least cost 6 on operators modifying \( A \) or \( B \).

\[
\begin{align*}
6 &= h\{A,B\} & \leq & \text{type-}A & + & \text{type-}B \\
6 &= h\{A,C\} & \leq & \text{type-}A & + & \text{type-}C \\
6 &= h\{B,C\} & \leq & \text{type-}B & + & \text{type-}C \\
18 & \leq 2\text{type-}A & + & 2\text{type-}B & + & 2\text{type-}C \\
9 & \leq \text{type-}A & + & \text{type-}B & + & \text{type-}C
\end{align*}
\]

⇒ at least cost 9 in any plan

Can we generalize this kind of reasoning?
Post-hoc optimization heuristic: linear program

Construct **linear program** for pattern collection $\mathcal{C}$:

- Variable $X_o$ for each operator $o \in \mathcal{O}$
  - Cost incurred by operator $o$ in a plan
  - $X_o \geq 0$ for each $o \in \mathcal{O}$
Construct **linear program** for pattern collection $C$:

- Variable $X_o$ for each operator $o \in O$
  - Cost incurred by operator $o$ in a plan
  - $X_o \geq 0$ for each $o \in O$
- PDB heuristics admissible

\[
h^P(s) \leq \sum_{o \in O} X_o \quad \text{for each pattern } P \in C
\]
Construct **linear program** for pattern collection $C$:

- Variable $X_o$ for each operator $o \in O$
  - Cost incurred by operator $o$ in a plan
  - $X_o \geq 0$ for each $o \in O$

- PDB heuristics admissible

\[ h^P(s) \leq \sum_{o \in O} X_o \text{ for each pattern } P \in C \]

- Can tighten constraints to

\[ h^P(s) \leq \sum_{o \in O: o \text{ affects } P} X_o \]
Post-hoc optimization heuristic: linear program

Construct **linear program** for pattern collection $\mathcal{C}$:

- Variable $X_o$ for each operator $o \in \mathcal{O}$
  - Cost incurred by operator $o$ in a plan
  - $X_o \geq 0$ for each $o \in \mathcal{O}$
- PDB heuristics admissible
  \[ h^P(s) \leq \sum_{o \in \mathcal{O}} X_o \text{ for each pattern } P \in \mathcal{C} \]
- Can tighten constraints to
  \[ h^P(s) \leq \sum_{o \in \mathcal{O} : o \text{ affects } P} X_o \]
- Total cost of the plan is $\sum_{o \in \mathcal{O}} X_o$
- **Minimizing total cost** leads to admissible estimate
Post-hoc optimization heuristic: definition and admissibility

**Definition (Post-hoc optimization heuristic)**

The post-hoc optimization heuristic $h^\text{PhO}_C$ is the objective value of the linear program for $C$ as described above.
Post-hoc optimization heuristic: definition and admissibility

Definition (Post-hoc optimization heuristic)

The post-hoc optimization heuristic $h^\text{PhO}_C$ is the objective value of the linear program for $C$ as described above.

Theorem

The post-hoc optimization heuristic is admissible.
Aside: cost partitioning

- Alternative way of using multiple patterns: operator cost partitioning
- Account only for a fraction of the actual operator costs in each PDB so that estimates can be summed up admissibly
Aside: cost partitioning

- Alternative way of using multiple patterns: operator cost partitioning
- Account only for a fraction of the actual operator costs in each PDB so that estimates can be summed up admissibly

Examples

- **Uniform cost partitioning**
  - Operator $o$ affects $k$ patterns $\Rightarrow$ each PDB uses cost $c(o)/k$

- **Optimal cost partitioning** [Katz and Domshlak]
  - Best way to admissibly partition costs
  - State-specific LP
Duality Theorem

- Rewrite minimization LP as maximization LP
- Same objective value
- Different view on the same problem

Dual of LP in $h_{\text{PhO}}$

- State-specific cost partitioning
- Scales operator costs for heuristic $h^P$ by a factor $Y_P$
- Much smaller than LP for optimal cost partitioning
Consider the dual of the LP solved by $h_{\text{PhO}}^C$ in state $s$. If we restrict the variables to integers, the objective value is the canonical heuristic value $h^C(s)$.\[\]
## Relation to canonical heuristic

### Theorem

Consider the **dual** of the LP solved by $h_{PhO}^C$ in state $s$. If we restrict the variables to integers, the objective value is the canonical heuristic value $h^C(s)$.

### Theorem

The post-hoc optimization heuristic $h_{PhO}^C$ **dominates** the canonical heuristic $h^C$. 
## Experimental results I

<table>
<thead>
<tr>
<th>Coverage</th>
<th>iPDB hill-climbing</th>
<th>Systematic (size 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$h^C$</td>
<td>$h^{\text{PhO}}$</td>
</tr>
<tr>
<td>IPC 2011</td>
<td>133</td>
<td>133</td>
</tr>
<tr>
<td>(280)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPC 1998–2008</td>
<td>456</td>
<td>459</td>
</tr>
<tr>
<td>(1116)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td>589</td>
<td>592</td>
</tr>
<tr>
<td>(1396)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

More detailed results on the poster.
Experimental results II

Why is $h^{PhO}$ better than $h^C$?

Additional evaluation on systematic pattern collections:
- Theoretical dominance of $h^{PhO}$?
- Faster computation?
Experimental results II

Why is $h_{PhO}$ better than $h_C$?

Additional evaluation on systematic pattern collections:
- Theoretical dominance of $h_{PhO}$?
  - Better guidance on only a few domains
- Faster computation?
Experimental results II

Why is $h^{PhO}$ better than $h^C$?

Additional evaluation on systematic pattern collections:

- Theoretical dominance of $h^{PhO}$?
  - Better guidance on only a few domains
- Faster computation?
  - Considered tasks solved by $h^{PhO}$ but not by $h^C$
  - Most ran out of memory during generation of $MAS(C)$
  - On these tasks, $h^C$ would be extremely slow
  - On commonly solved tasks $h^C$ tends to be faster
Experimental results III

Time to calculate heuristic value of initial state

Colored by $|C|$
Conclusion

Two contributions

- **Post-hoc optimization heuristic**
  - Middle ground between canonical heuristic and optimal cost partitioning

- **Systematic generation of interesting patterns**
  - Improves over iPDB hill climbing when used with suitable heuristic
Thank you for your attention!

Poster presentation

- Friday 8:30 – 9:45
### $h^{\text{PhO}}$ LP

Minimize $\sum_{o \in O} X_o$ subject to

$$\sum_{o \in O: o \text{ affects } P} X_o \geq h^P(s) \quad \text{for all } P \in C$$

$$X_o \geq 0 \quad \text{for all } o \in O.$$

### Corresponding dual program to $h^{\text{PhO}}$ LP

Maximize $\sum_{P \in C} Y_P h^P(s)$ subject to

$$\sum_{P \in C: o \text{ affects } P} Y_P \leq 1 \quad \text{for all } o \in O$$

$$Y_P \geq 0 \quad \text{for all } P \in C.$$
Post-hoc optimization heuristic: simplifying the LP

Reduce size of LP

- **Aggregate variables** which always occur together in constraints
- Happens when several operators are **relevant for exactly the same PDBs**
- Merge individual variables into one new variable
  - Represents their sum

**Example task**

Merged all operators modifying $A$ into variable *type-A*
## Detailed experimental results I

<table>
<thead>
<tr>
<th>Problem</th>
<th>$h^C$</th>
<th>$HC^C$</th>
<th>$h^{PhO}$</th>
<th>$h^{OCP}$</th>
<th>$HC^{PhO}$</th>
<th>$h^{PhO}$</th>
<th>$h^{OCP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>barman (20)</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>elevators (20)</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td>16</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>floortile (20)</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>nomystery (20)</td>
<td>16</td>
<td>16</td>
<td>3</td>
<td>16</td>
<td>18</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>openstacks (20)</td>
<td>14</td>
<td>14</td>
<td>5</td>
<td>14</td>
<td>5</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>parcprinter (20)</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>parking (20)</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pegsol (20)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>scanalyzer (20)</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>sokoban (20)</td>
<td>20</td>
<td>20</td>
<td>18</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>tidybot (20)</td>
<td>14</td>
<td>14</td>
<td>6</td>
<td>11</td>
<td>14</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>transport (20)</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>visitall (20)</td>
<td>16</td>
<td>16</td>
<td>10</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>woodworking (20)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td><strong>Sum IPC 2011</strong> (280)</td>
<td>133</td>
<td>133</td>
<td>56</td>
<td>122</td>
<td>126</td>
<td>158</td>
<td>43</td>
</tr>
<tr>
<td><strong>IPC 1998–2008</strong> (1116)</td>
<td>456</td>
<td>459</td>
<td>241</td>
<td>426</td>
<td>446</td>
<td>475</td>
<td>231</td>
</tr>
<tr>
<td><strong>Sum</strong> (1396)</td>
<td>589</td>
<td>592</td>
<td>297</td>
<td>548</td>
<td>572</td>
<td>633</td>
<td>274</td>
</tr>
</tbody>
</table>
## Detailed experimental results II

<table>
<thead>
<tr>
<th></th>
<th>Sys(^1)</th>
<th></th>
<th>Sys(^2)</th>
<th></th>
<th>Sys(^3)</th>
<th></th>
<th>Sys(^*)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(h^C)</td>
<td>(h^{PhO})</td>
<td>(h^C)</td>
<td>(h^{PhO})</td>
<td>(h^C)</td>
<td>(h^{PhO})</td>
<td>(h^{PhO})</td>
</tr>
<tr>
<td>barman (20)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>elevators (20)</td>
<td>9</td>
<td>9</td>
<td>16</td>
<td>15</td>
<td>16</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>floortile (20)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>nomystery (20)</td>
<td>12</td>
<td>12</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>openstacks (20)</td>
<td>14</td>
<td>14</td>
<td>5</td>
<td>14</td>
<td>2</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>parcprinter (20)</td>
<td>11</td>
<td>11</td>
<td>7</td>
<td>13</td>
<td>5</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>parking (20)</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>pegsol (20)</td>
<td>17</td>
<td>17</td>
<td>5</td>
<td>17</td>
<td>1</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>scanalyzer (20)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>sokoban (20)</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>tidybot (20)</td>
<td>13</td>
<td>13</td>
<td>14</td>
<td>14</td>
<td>8</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>transport (20)</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>visitall (20)</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>woodworking (20)</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>10</td>
<td>1</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td><strong>Sum IPC 2011</strong></td>
<td><strong>143</strong></td>
<td><strong>143</strong></td>
<td><strong>126</strong></td>
<td><strong>158</strong></td>
<td><strong>109</strong></td>
<td><strong>142</strong></td>
<td><strong>174</strong></td>
</tr>
<tr>
<td><strong>IPC 1998–2008</strong></td>
<td><strong>449</strong></td>
<td><strong>449</strong></td>
<td><strong>446</strong></td>
<td><strong>475</strong></td>
<td><strong>406</strong></td>
<td><strong>422</strong></td>
<td><strong>501</strong></td>
</tr>
<tr>
<td><strong>Sum (1396)</strong></td>
<td><strong>592</strong></td>
<td><strong>592</strong></td>
<td><strong>572</strong></td>
<td><strong>633</strong></td>
<td><strong>515</strong></td>
<td><strong>564</strong></td>
<td><strong>675</strong></td>
</tr>
</tbody>
</table>