An Admissible HTN Planning Heuristic

Pascal Bercher and Gregor Behnke and Daniel Höller and Susanne Biundo

Institute of Artificial Intelligence

August, 23th, 2017
We are concerned with solving hierarchical planning problems.

- How to solve these problems based on well-informed domain-independent heuristics?
- How to provide optimality guarantees?
We are concerned with solving hierarchical planning problems.

- How to solve these problems based on well-informed domain-independent heuristics?
- How to provide optimality guarantees?

We base our formalization upon *hybrid planning*, which fuses

- hierarchical task network (HTN) planning with
- partial order causal link (POCL) planning.
Related Work:

HTN planners using control knowledge (e.g., SHOP2).
Hierarchical heuristic search planners for different problem classes (e.g., hybrid, HGN, GTN, HTN+preferences).
Related Work:

- HTN planners using control knowledge (e.g., SHOP2).
- Hierarchical heuristic search planners for different problem classes (e.g., hybrid, HGN, GTN, HTN+preferences).
Related Work:

- HTN planners using control knowledge (e.g., SHOP2).
Related Work:

- HTN planners using control knowledge (e.g., SHOP2).
- Hierarchical heuristic search planners for different problem classes (e.g., hybrid, HGN, GTN, HTN+preferences).
A hybrid planning problem is given in terms of:

- a planning domain stating the tasks (primitive and abstract) and methods (how to refine an abstract action?), and
- an initial partial plan, which needs to be refined into a solution.
A hybrid planning problem is given in terms of:

- a planning domain stating the tasks (primitive and abstract) and methods (how to refine an abstract action?), and
- an initial partial plan, which needs to be refined into a solution.

A solution is a partially ordered plan, such that:

- it is a primitive refinement of the initial partial plan,
- all its linearizations are executable in the initial state, and
- all its linearizations satisfy the goal description.

→ see example given next.
An Admissible HTN Planning Heuristic

August, 23th, 2017

4 / 11
An Admissible HTN Planning Heuristic

August, 23th, 2017
An Admissible HTN Planning Heuristic

August, 23th, 2017 4 / 11
An Admissible HTN Planning Heuristic

August, 23th, 2017

4 / 11
An Admissible HTN Planning Heuristic
An Admissible HTN Planning Heuristic
related work
formal framework
admissible heuristic
empirical results
summary

Init

\text{signal}(BR, \text{AUDIO})
\quad \text{plugIn}(BR, CINCH, \text{AUDIO})
\quad \text{signal}(CINCH, \text{AUDIO})
\quad \text{connected}(BR, CINCH)

\text{signal}(CINCH, \text{AUDIO})
\quad \text{connected}(BR, CINCH)
\quad \text{connected}(CINCH, AV-Rec)

\text{signal}(CINCH, \text{AUDIO})
\quad \text{connected}(CINCH, AV-Rec)
\quad \text{signal}(AV-Rec, \text{AUDIO})

\text{setup}(?dev_2, TV, \text{VIDEO})
\quad \text{signal}(TV, \text{VIDEO})
\quad \text{signal}(TV, \text{VIDEO})

Goal

An Admissible HTN Planning Heuristic
**An Admissible HTN Planning Heuristic**

**August, 23th, 2017**
An Admissible HTN Planning Heuristic
**An Admissible HTN Planning Heuristic**

August, 23th, 2017

4 / 11
An Admissible HTN Planning Heuristic

August, 23th, 2017
How to come up with heuristic functions?

- Both HTN and hybrid problems are \textit{undecidable}, we hence need severe problem relaxations that them \textit{decidable} and \textit{tractable}.
- \(\Rightarrow\) We allow task insertion.
- \(\Rightarrow\) We make the problem acyclic.
How to come up with heuristic functions?

- Both HTN and hybrid problems are *undecidable*, we hence need severe problem relaxations that them *decidable* and *tractable*.

- ↳ We allow task insertion.

- ↳ We make the problem acyclic.
How to come up with heuristic functions?

- Both HTN and hybrid problems are *undecidable*, we hence need severe problem relaxations that them *decidable* and *tractable*.

- \( \Rightarrow \) We allow task insertion.

- \( \Rightarrow \) We make the problem acyclic.
How does the heuristic work?

**Step 1:**
- Build a so-called task-decomposition graph (TDG), which represents the hierarchical problem structure.

**Step 2:**
- Calculate heuristic estimates for all tasks in that TDG.

**Step 3:**
- For a given plan, retrieve the TDG’s estimates for all abstract tasks in that plan.
Task decomposition Graph (TDG), example:

A TDG is a ground representation of the task hierarchy.

It is a (possibly cyclic) bipartite graph \( \langle V_T, V_M, E_{T \rightarrow M}, E_{M \rightarrow T} \rangle \) consisting of:

- task vertices \( V_T \) (abstract task vertices are OR nodes)
- method vertices \( V_M \) (which are AND nodes)
Let $\langle V_T, V_M, E_{T\rightarrow M}, E_{M\rightarrow T} \rangle$ be a ground TDG. Then, we can calculate a heuristics by exploiting its AND/OR structure:

$$h_T(v_t) := \begin{cases} \text{cost}(v_t) & \text{if } v_t \text{ is primitive} \\ \min_{(v_t, v_m) \in E_{T\rightarrow M}} h_M(v_m) & \text{else} \end{cases}$$

For a method vertex $v_m = \langle PS, \prec, CL, VC \rangle$, we set:

$$h_M(v_m) := \sum_{(v_m, v_t) \in E_{M\rightarrow T}} h_T(v_t)$$
Heuristic computation, example:

\[ h(m'_{t_3}) = \text{cost}(t_4(c_1)) \]
\[ h(m'_{t_3'}) = \text{cost}(t_4(c_1)) \]
\[ + \text{cost}(t_5(c_1)) \]
\[ + \text{cost}(t_2(c_1)) \]
\[ h(t_3(c_1)) = \min\{ h(m'_{t_3}), h(m'_{t_3'}) \} \]

\[ \ldots \]
Heuristic computation, example:

\[
\begin{align*}
  h(m_{t_3}') &= \text{cost}(t_4(c_1)) \\
  h(m_{t_3}') &= \text{cost}(t_4(c_1)) + \text{cost}(t_5(c_1)) \\
  h(t_3(c_1)) &= \min \{ h(m_{t_3}'), h(m_{t_3}') \} \\
  \ldots
\end{align*}
\]
Heuristic computation, example:

\[
\begin{align*}
h(m'_{t_3}) &= \text{cost}(t_4(c_1)) \\
h(m'_t) &= \text{cost}(t_4(c_1)) \\
&\quad + \text{cost}(t_5(c_1)) \\
&\quad + \text{cost}(t_2(c_1)) \\
h(t_3(c_1)) &= \min\{h(m'_{t_3}), h(m'_t)\} \\
&\quad \ldots
\end{align*}
\]
Heuristic computation, example:

\[ h(m'_t) = \text{cost}(t_4(c_1)) \]
\[ h(m'_t') = \text{cost}(t_4(c_1)) \]
\[ + \text{cost}(t_5(c_1)) \]
\[ + \text{cost}(t_2(c_1)) \]
\[ h(t_3(c_1)) = \min\{h(m'_t), h(m'_t')\} \]

...
<table>
<thead>
<tr>
<th>Strategy</th>
<th>UM-Tr. (21 inst.)</th>
<th>SmartPh. (5 inst.)</th>
<th>Satellite (22 inst.)</th>
<th>Woodw. (11 inst.)</th>
<th>Summary (59 inst.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#s</td>
<td>#o</td>
<td>cost</td>
<td>#s</td>
<td>#o</td>
</tr>
<tr>
<td><strong>blind</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniform</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>BF</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DF</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td><strong>UMCP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BF</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>DF</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Compile</td>
<td>18</td>
<td>18</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Compileopt</td>
<td>16</td>
<td>16</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td><strong>A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>ADD-r</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Relax</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>OC</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>TDGm/-rec</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>TDGc/-rec</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td><strong>Greedy-A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>ADD-r</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Relax</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>OC</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>TDGm/-rec</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>TDGc</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>TDGc-rec</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Summary:

- Introduced the first admissible heuristic for standard HTN and hybrid planning.
- The proposed heuristic(s) perform best both in terms of coverage and plan quality.

Also in the paper and poster:

- A variant of the heuristic tailored to hybrid planning systems.
- Investigation of TDG recomputation to improve heuristic accuracy.