Learning Heuristic Selection with Dynamic Algorithm Configuration

David Speck¹, André Biedenkapp¹, Frank Hutter¹,², Robert Mattmüller¹ and Marius Lindauer³
(speckd, biedenka, fh, mattmuel)@informatik.uni-freiburg.de, lindauer@tnt.uni-hannover.de
¹University of Freiburg, ²Bosch Center for Artificial Intelligence, ³Leibniz University Hannover

Motivation

What state should I expand next?

State $s_i$

$h_1$

$h_2$

$h_3$

$h_4$

I will tell you from my experience!

Who is correct?

Planner

RL Agent

Dynamic Algorithm Configuration – Theoretical Properties

▶ An optimal DAC policy is at least as good as an optimal AS policy and an optimal AAC policy.

▶ There is a family of planning tasks so that a DAC policy expands exponentially fewer states until a plan is found.

Features and Rewards

▶ Features for each heuristic $h \in H$ (open list)
  ▶ $\max_h, \min_h, \mu_h, \sigma^2_h, \#_h$ and $t \in \mathbb{N}_0$
  ▶ Difference of each feature between $t-1$ and $t$
  ▶ Reward: $-1$ for each expansion step until solution is found

Experiments

▶ $H = \{h_{\text{ff}}, h_{\text{cg}}, h_{\text{cea}}, h_{\text{add}}\}$

▶ 6 domains with 100 instances per train/test Set

▶ ε-greedy deep Q-learning (double DQN)

▶ 2-layer network with 75 hidden units

▶ 5 different DAC policies per domain

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CONTROL POLICY</th>
<th>SINGLE HEURISTIC</th>
<th>BEST AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain (# Inst.)</td>
<td>RL</td>
<td>RND</td>
<td>ALT</td>
</tr>
<tr>
<td>BARMAN (100)</td>
<td>84.4</td>
<td>83.8</td>
<td>83.3</td>
</tr>
<tr>
<td>BLOCKS (100)</td>
<td>92.9</td>
<td>83.6</td>
<td>83.7</td>
</tr>
<tr>
<td>CHILD (100)</td>
<td>88.0</td>
<td>86.2</td>
<td>86.7</td>
</tr>
<tr>
<td>ROVERS (100)</td>
<td>95.2</td>
<td>96.0</td>
<td>96.0</td>
</tr>
<tr>
<td>SOKOBAN (100)</td>
<td>87.7</td>
<td>87.1</td>
<td>87.0</td>
</tr>
<tr>
<td>VISITALL (100)</td>
<td>56.9</td>
<td>51.0</td>
<td>51.5</td>
</tr>
<tr>
<td>SUM (600)</td>
<td>505.1</td>
<td>487.7</td>
<td>488.2</td>
</tr>
</tbody>
</table>

▶ Our approach based on RL performs overall best

▶ Best Algorithm Selection (Oracle) is worse than control policies

Conclusion and Future Work

DAC can improve heuristic selection.

▶ Considers instance, time step and planner state

▶ Can improve search performance exponentially

▶ It is possible to learn good policies

▶ Future: Investigate domain-specific state features