Expressing and Exploiting the Common Subgoal Structure of Classical Planning Domains Using Sketches

Dominik Drexler, Jendrik Seipp, Hector Geffner

1Linköping University, Linköping, Sweden, 2ICREA & Universitat Pompeu Fabra, Barcelona, Spain
{dominik.drexler, jendrik.seipp}@liu.se, hector.geffner@upf.edu

In a Nutshell

- Classical planning
- We consider tractable planning domains
- Policy sketch defines subgoal structure
- Contribution: We encode subgoal structure using compact policy sketches to solve whole domains in provably low poly time
- Subproblems are solved with iterated width
- Partial plans are serialized

Iterated Width

- IW(k) is breadth-first search where a newly generated state s is pruned if novelty(s) > k
- novelty(s) := smallest size of tuple of atoms made true for the first time
- IW(k) requires \( \exp(k) \) time
- When does IW(k) solve a problem?

The Problem of Unbounded Width

- Single goal atom \( \Rightarrow \) often small width
- Conjunctive goals \( \Rightarrow \) often unbounded width
- Serialized Iterated Width (SIW)
  - SIW(k) runs sequence of IW(\( k \)) searches
  - Each IW search decreases goal counter \#g
  - Subproblems of achieving single goal atom
- SIW still fails if ...
  - it traps into an unsolvable state
  - it generates a subproblem of greater width
  - the subproblem has too large width
- Richer decompositions using policy sketches

Policy Sketches

- Consider some possibly infinite class of problems \( Q \) over some common domain \( D \)
- Policy sketch (sketch) \( R \) defines subgoal structure in every \( P \in Q \)
- Sketch \( R \) is set of rules of form \( C \Rightarrow E \) over features \( \Phi \)
- Sketch width \( w_R(Q) \) is maximum width of all subproblems in all \( P \in Q \)
- SIWR serializes according to subgoals defined by sketch \( R \)
- Theorem: if \( w_R(Q) \leq k \) then SIWR solves all \( P \in Q \) in \( \exp(k) \) time

Example Domain: Grid

- Domain description:
  - Robot, key(s), lock(s) distributed in a grid
  - Robot cannot move on a place with closed lock
  - Goal: well place keys; can require opening locks
  - SIW generates subproblem of large width
- Features \( \Phi = \{ l, \#g, kl, kg \} \)
  - \( l \) is number of closed locks
  - \( \#g \) is number of wellplaced keys
  - \( kl \) whether robot holds key to open lock
  - \( kg \) whether robot holds misplaced key
- Rules \( R_\Phi = \{ r_1, r_2, r_3, r_4 \} \)
  - \( r_1 = \{ l > 0 \} \Rightarrow \{ l, \#g, kl, kg, \#g \} \)
  - \( r_2 = \{ l = 0, \#g > 0 \} \Rightarrow \{ \#g, kl, kg \} \)
  - \( r_3 = \{ l > 0, \neg kl \} \Rightarrow \{ kl, kg \} \)
  - \( r_4 = \{ l = 0, \#g > 0, \neg kg \} \Rightarrow \{ kl, kg \} \)
  - \( w_{R_\Phi}(Q) = 2 \) for \( R_1 = \{ r_1, r_3 \} \)
  - \( w_{R_\Phi}(Q) = 1 \) for \( R_2 = \{ r_1, \ldots, r_4 \} \)

Experiments

<table>
<thead>
<tr>
<th>Domain</th>
<th>SIW(2)</th>
<th>SIW(3)</th>
<th>LAMA</th>
<th>Dual-BFWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barman (40)</td>
<td>S</td>
<td>T</td>
<td>MW</td>
<td>S</td>
</tr>
<tr>
<td>Childs (40)</td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>40</td>
</tr>
<tr>
<td>Driverlog (20)</td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>20</td>
</tr>
<tr>
<td>Floortile (20)</td>
<td>0</td>
<td>–</td>
<td>–</td>
<td>20</td>
</tr>
<tr>
<td>Grid (5)</td>
<td>0</td>
<td>1</td>
<td>0.1</td>
<td>5</td>
</tr>
<tr>
<td>Schedule (150)</td>
<td>62</td>
<td>1349.1</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>TPP (30)</td>
<td>11</td>
<td>74.7</td>
<td>2</td>
<td>30</td>
</tr>
</tbody>
</table>

# Solved 0/7 7/7 5/7 4/7