**End-to-End Classical Planning using CP and Belief Propagation**

Damien Van Meerbeeck, Gilles Pesant, and Hendrik Seipp

1. Machine Reasoning Lab, Linköping University, Sweden
2. Quossé Lab, Polytechnique Montréal, Canada

---

**Classical Planning**

- **Problem**: Finding a sequence of actions transforming an initial situation into one that satisfies a goal condition.
- **Solution**: Use CPBP solvers to solve planning tasks using *Constraint Programming* (CP).
- **Advantages**: Let users specify additional constraints on the plan in a declarative way; use the power of CPBP solvers to scale better for large planning tasks.

**Input planning task**

Planning task in first order planning definition language (PDDL):
- **Objects**
- **Predicates**
- **Initial and goal states**
- **Actions** with preconditions and effects on predicates

**Ground finite-domain task**

Use the Fast Downward planning system to translate the input task into a ground finite-domain task (SAS*).

**Automata**

Project the task to each variable of the SAS* task, to obtain a Factored Transition System. Additionally, group parallel actions (Group) and prune irrelevant actions (Group+Prune).

**CP Model**

Model of the planning task in CP:
- **Plan**: An array of integer variables with a fixed plan length.
- **Task Variables**: Regular constraint for each task variable to enforce their domain task.

**Search**

Solve using MiniCPBP solver with belief propagation and the maximum marginal branching heuristic on the actions of the plan.

---

**Preliminary Results**

<table>
<thead>
<tr>
<th>Planning Domain</th>
<th>Outcome</th>
<th>NoGroup</th>
<th>Group</th>
<th>Group+Prune</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miconic (150)</td>
<td>solved opt.</td>
<td>38</td>
<td>42</td>
<td>41</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>out of memory</td>
<td>60</td>
<td>28</td>
<td>28</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>out of time</td>
<td>52</td>
<td>80</td>
<td>81</td>
<td>101</td>
</tr>
<tr>
<td>Scanalyzer (41)</td>
<td>solved (solved)</td>
<td>5 (11)</td>
<td>5 (15)</td>
<td>5 (21)</td>
<td>(33) 33</td>
</tr>
<tr>
<td></td>
<td>out of memory</td>
<td>18</td>
<td>18</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>out of time</td>
<td>18</td>
<td>18</td>
<td>24</td>
<td>8</td>
</tr>
</tbody>
</table>

**Future Works**

- Merging automata
- Action Space reduction
- New constraints (landmarks, operator counting, ...)

---

**Motivation**

- Solve planning tasks using *Constraint Programming* (CP).
- Let users specify additional constraints on the plan in a declarative way.
- Use the power of CPBP solvers to scale better for large planning tasks.