Extending Graph Neural Networks for Generalized Stochastic Planning

Ziqi Zhang    Florian Geißer

The Australian National University
Computing General Policies for Planning Problems

Policy:

• Mapping from states to actions
• Solution to a specific planning problem

General Policy:

• Applicable to every problem of a given domain class
• Policies commonly represented as Neural Networks
Different architectures support different modeling languages

- PDDL (classical planning / SSPs):
  - STRIPS Hypergraph Networks (Shen et al. 2020)
  - Action Schema Networks (Toyer et al. 2018)
- RDDL (MDPs)
  - SymNet (Garg et al. 2020)
  - TrapsNet (Garg et al. 2019)
Our work: A novel network architecture for planning

- builds on the ideas of TrapsNet
  - but allows fluents of arbitrary size
- is applied to RDDL problems
  - but is in principle independent of the modeling language
- allows to train a graph network on small instances
- evaluation on IPC 2014 domains shows promising results
A Relational Markov Decision Process consists of:

- A set of types $T$
- State, action, and static predicate symbols $F$
  - Denote relations of and between typed objects
- A lifted transition function $P$ and reward function $R$
  - Describe the dynamics of the first-order MDP

Given a set of objects, applying predicates to type-consistent objects is known as the process of grounding and yields a factored MDP.
Running Example: Elevator domain

- Types: floor, elevator
- State predicates:
  - person-waiting-at-floor(elevator, floor)
  - elevator-at-floor(elevator, floor)
- Action predicates:
  - go-up(elevator), go-down(elevator)
  - open-door(elevator), close-door(elevator)
- Static predicates:
  - TOP-FLOOR(floor)
  - BOTTOM-FLOOR(floor)
  - ADJACENT-FLOORS(floor, floor)
A domain graph shows relations between types

Domain Graph
An instance graph shows **relations between objects**

**Domain Graph**

- `<floor>`
  - BOTTOM-FLOOR `<floor>`
  - TOP-FLOOR `<floor>`

**Instance Graph**

- `<f0>`
  - BOTTOM-FLOOR `<f0>`
  - TOP-FLOOR `<f0>`

- `<f1>`
  - BOTTOM-FLOOR `<f1>`
  - TOP-FLOOR `<f1>`

- `<f2>`
  - BOTTOM-FLOOR `<f2>`
  - TOP-FLOOR `<f2>`

- `<f0,f1>`
  - ADJACENT-FLOORS `<f0,f1>`

- `<f1,f2>`
  - ADJACENT-FLOORS `<f1,f2>`
Network Architecture

- Domain/Instance are used in a graph neural network
- Every instance graph vertex $v$ has an initial embedding $h_v^0$
- Embedding at step $k + 1$ is computed by forward pass over the network at step $k$
  - we use the attention mechanism to aggregate a set of vectors with unknown cardinality
- Final policy is computed by action decoder as a distribution over grounded actions
Empirical Evaluation

• Four domains of the IPPC 2014
• Network trained on the three smallest instances
• We compare against PROST (Keller et al. 2012)
• Our empirical evaluation is only a preliminary study
### Evaluation

<table>
<thead>
<tr>
<th>Instance</th>
<th>Tamarisk domain</th>
<th>Wildfire domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>THTS</td>
<td>Network</td>
</tr>
<tr>
<td>1 (trained on)</td>
<td>-151±19.7</td>
<td>-146±15.4</td>
</tr>
<tr>
<td>2 (trained on)</td>
<td>-542±27.7</td>
<td>-530±24.2</td>
</tr>
<tr>
<td>3 (trained on)</td>
<td>-206±27.3</td>
<td>-222±23.8</td>
</tr>
<tr>
<td>4</td>
<td>-826±28.4</td>
<td>-822±28.6</td>
</tr>
<tr>
<td>5</td>
<td>-723.9±41.4</td>
<td>-655±38</td>
</tr>
<tr>
<td>6</td>
<td>-1071±26.2</td>
<td>-1045±33.2</td>
</tr>
<tr>
<td>7</td>
<td>-891±43.2</td>
<td>-823±41.7</td>
</tr>
<tr>
<td>8</td>
<td>-1285±23.5</td>
<td>-1251±28.2</td>
</tr>
<tr>
<td>9</td>
<td>-902±53.2</td>
<td>-860±59.9</td>
</tr>
<tr>
<td>10</td>
<td>-1346±36.3</td>
<td>-1290±39.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Elevator domain</th>
<th>Sysadmin domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>THTS</td>
</tr>
<tr>
<td>1 (trained on)</td>
<td>-42.5±2.5</td>
</tr>
<tr>
<td>2 (trained on)</td>
<td>-23.8±2.2</td>
</tr>
<tr>
<td>3 (trained on)</td>
<td>-61.6±1.9</td>
</tr>
<tr>
<td>4</td>
<td>-54.2±4.2</td>
</tr>
<tr>
<td>5</td>
<td>-64.9±4.6</td>
</tr>
<tr>
<td>6</td>
<td>-83.3±3.8</td>
</tr>
<tr>
<td>7</td>
<td>-79.4±5.4</td>
</tr>
<tr>
<td>8</td>
<td>-88.2±5.2</td>
</tr>
<tr>
<td>9</td>
<td>-107.5±5.4</td>
</tr>
<tr>
<td>10</td>
<td>-66.5±5.9</td>
</tr>
</tbody>
</table>
Conclusion

- **Novel network architecture** based on domain graphs
- Allows for problems with **arbitrary predicate size**
- **Generalizes well across instances** of different size
- Interesting aspect: network is in principle independent of the modeling language used


Z. Zhang, F. Geißer – Extending Graph Neural Networks for Generalized Stochastic Planning