Graph Learning for Numeric Planning

Dillon Z. Chen Sylvie Thiébaux

LAAS CNRS

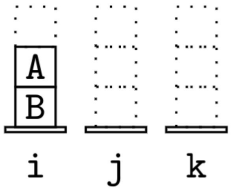
Australian National University

NeurIPS 2024
Vancouver, Canada

What is Numeric Planning?

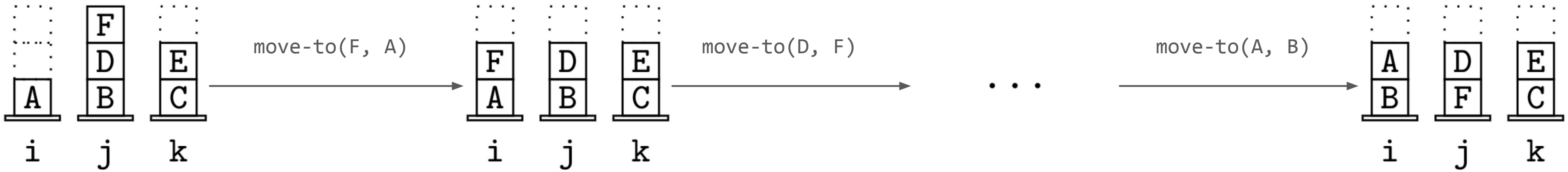
- goal-conditioned sequential decision making
- factored representations of states
    - predicates for Boolean values
    - functions for numeric values
- e.g. Capacity Constrained Blocksworld

goal condition



initial state

goal state




```
on(A, i)
on(F, D), on(D, B), on(B, j)
on(E, C), on(C, k)
capacity(i) = 2, capacity(j) = 0, capacity(k) = 1
```

```
on(F, A), on(A, i)
on(D, B), on(B, j)
on(E, C), on(C, k)
capacity(i) = 1, capacity(j) = 1, capacity(k) = 1
```

```
on(A, B), on(B, i)
on(D, F), on(F, j)
on(E, C), on(C, k)
capacity(i) = 1, capacity(j) = 1, capacity(k) = 1
```

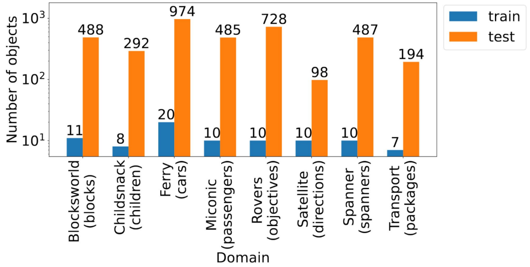
Planning and RL: What’s the Difference?

Both solve MDPs!

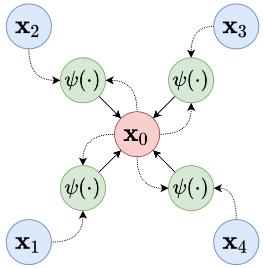
Planning	Reinforcement Learning
model-known	model-free or model-based
goal-conditioned, minimise cost	maximise reward
transitions modelled symbolically	transitions modelled as distributions
search algorithms: A*, GBFS, iLAO*, LRTDP	search algorithms: MTCS, UCT, TD(λ)
heuristic functions (cost-to-go estimator)	value functions (expected reward)
algorithms guided by models	algorithms guided by rewards

Learning for Planning

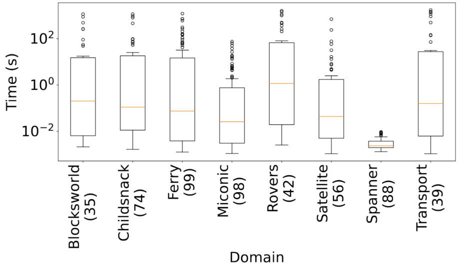
- generalise across object sizes → out-of-distribution learning



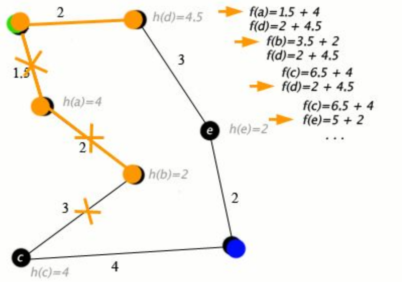
- graph learning for relational data and arbitrarily sized inputs



- supervised learning from quick to generate training labels



- learn heuristic functions for heuristic search

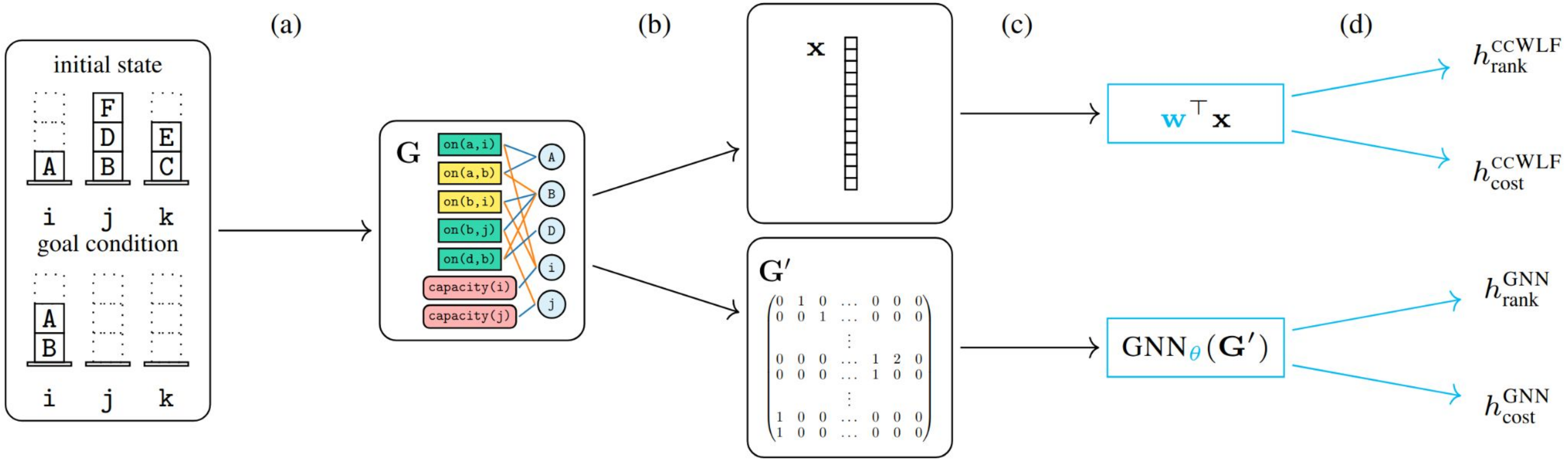


Learning Pipeline

(a) numeric planning tasks **transformed into graphs**

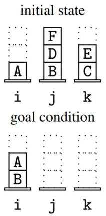
(b/c) graphs input into parameterised graph learning models

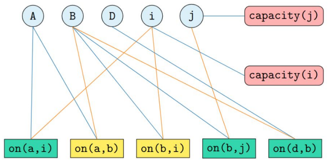
(d) models **learn a heuristic function** as either cost-to-go estimators or ranking functions



New Contributions

- graph representation for numeric planning





$$
\begin{array}{l|ccccccc}
 & \text{OH(obj)} & \text{OH(on,a\_p\_n)} & \text{OH(on,a\_p\_g)} & \dots & \text{OH(capacity)} & r_1 = \text{num. value} & r_2 = \text{goal error} \\
\hline
\texttt{on(a, x)} & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\
\texttt{on(a, b)} & 0 & 0 & 1 & \dots & 0 & 0 & 0 \\
\dots & & & & \dots & & & \\
\texttt{capacity(i)} & 0 & 0 & 0 & \dots & 1 & 2 & 0 \\
\texttt{capacity(j)} & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\
\dots & & & & \dots & & & \\
\texttt{i} & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\
\texttt{j} & 1 & 0 & 0 & \dots & 0 & 0 & 0
\end{array}
$$

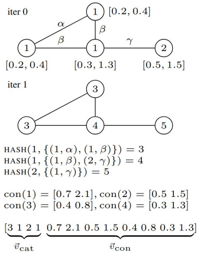
- graph kernel for categorical-continuous attributes

Algorithm 1: CCWL algorithm

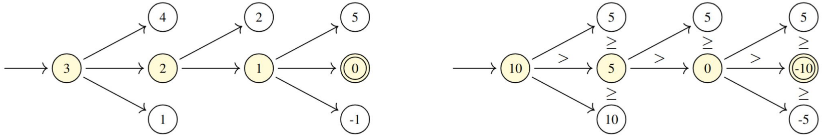
Data: A graph $\mathbf{G} = \langle \mathbf{V}, \mathbf{E}, \mathbf{F}_{\text{cat}}, \mathbf{F}_{\text{con}}, \mathbf{L} \rangle$ with continuous and categorical attributes, a deterministic HASH function, allowed colours $\mathbf{C} = [|\mathbf{C}|]$, a pooling function POOL, and number of CCWL iterations L.

Result: Feature vector of size $\mathbb{R}^{(1+d)|\mathbf{C}|}$.

$\kappa^0(v) \leftarrow \mathbf{F}_{\text{cat}}(v), \forall v \in \mathbf{V}$
for $j \in [L]$ **do for** $v \in \mathbf{V}$ **do**
  $\kappa^j(v) \leftarrow \text{HASH}\left(\kappa^{j-1}(v), \bigcup_{\iota \in \Sigma_{\text{E}}} \{(\kappa^{j-1}(u), \iota) \mid u \in \mathbf{N}_\iota(v)\}\right)$
$\mathbf{M} \leftarrow \bigcup_{j=0,\dots,L} \{\!\{\kappa^j(v) \mid v \in \mathbf{V}\}\!\}$
$\vec{v}_{\text{cat}} \leftarrow \left[\text{COUNT}(\mathbf{M}, c_1), \dots, \text{COUNT}(\mathbf{M}, c_{|\mathbf{C}|})\right]$
$\mathbf{S}_i = \left\{v \mid v \in \mathbf{V} \text{ s.t. } \exists j \in \{0\} \cup [L], \kappa^j(v) = c_i\right\}, \forall i \in \mathbf{C}$
$\vec{v}_{\text{con}} \leftarrow [\text{con}(1) \| \dots \| \text{con}(|\mathbf{C}|)],\ \text{con}(i) = \text{POOL}_{v \in \mathbf{S}_i}(\mathbf{F}_{\text{con}}(v))$
return $\vec{v}_{\text{cat}} \| \vec{v}_{\text{con}}$

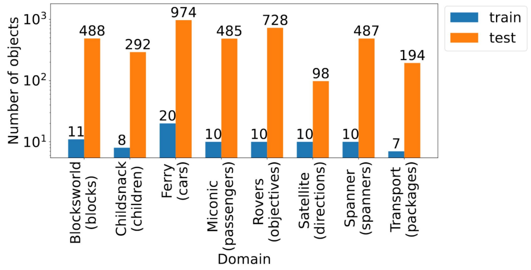


- LP ranking formulation for learning heuristic functions



Experiments

- train on labelled data from small training tasks



- evaluate with learned heuristic functions on large tasks
- competitive against numeric planners
    - purple = numeric planner
    - orange = learning planner

Higher values are better (↑)