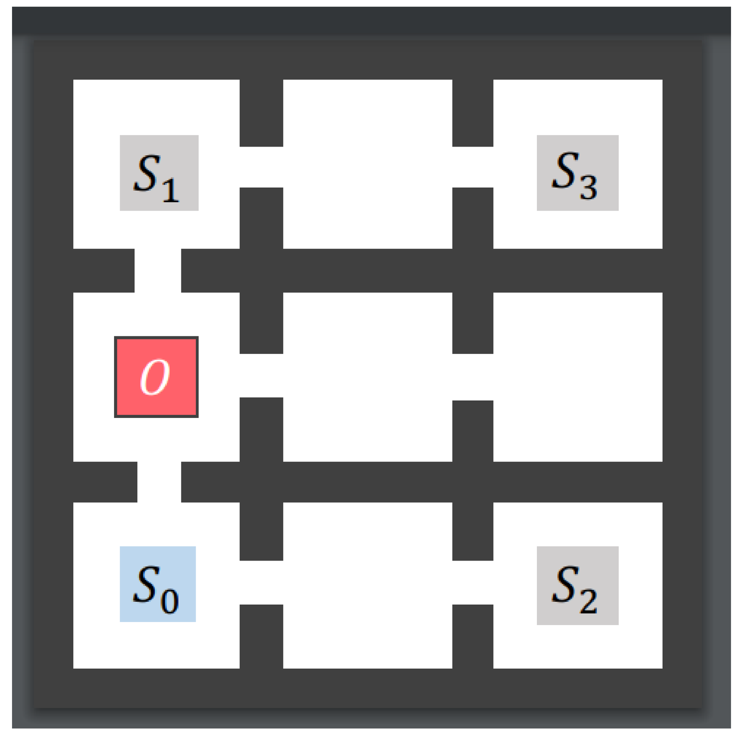
Compositional Reinforcement Learning from Logical Specifications

Kishor Jothimurugan, Suguman Bansal, Osbert Bastani and Rajeev Alur

Penn University of Pennsylvania



Task

Visit S_1 or S_2
then visit S_3
Always avoid O

Logical Specifications

- Hard to write well-shaped reward functions for complex tasks
  - We instead use logical specifications

$$\phi = (\textbf{choose}\ (\text{reach } S_1, \text{reach } S_2); \text{reach } S_3)\ \textbf{ensuring}\ \text{avoid } O$$

Problem Statement

Given an MDP M with unknown transition probabilities and a specification ϕ we want to compute a policy π^* such that

$$\pi^* \in \arg\max_{\pi} \Pr_{\zeta \sim D_{\pi}^{M}}[\zeta \vDash \phi]$$

Drawbacks of Existing Approaches

- Poor scalability w.r.t. complexity of specification
- Not designed to solve tasks that require high-level planning

Contributions

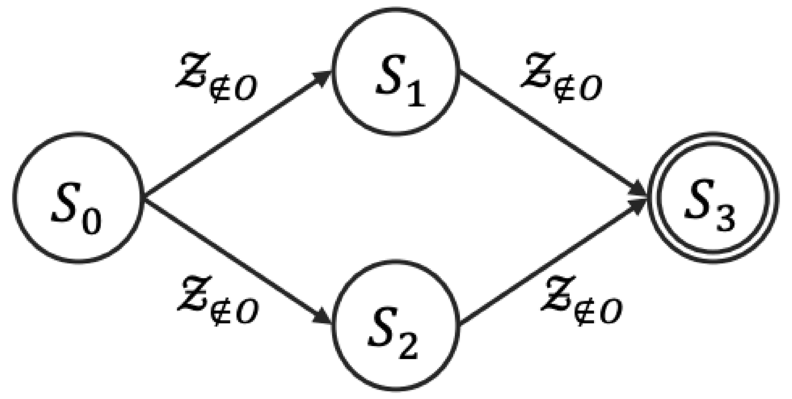
1. ***Compositional algorithm*** that **interleaves** high-level planning with low-level RL
2. ***Theoretical analysis*** showing that our algorithm's objective is a lower bound on satisfaction probability
3. ***Experimental evaluation*** on challenging benchmarks

Compositional Algorithm

Phase I: Construct Abstract Graph

An ***abstract graph*** is a DAG-like structure derived from the given specification (automatically)

- Vertices are ***sets of states***
- Edges are ***subtasks***
- Edge labels denote ***constraints***



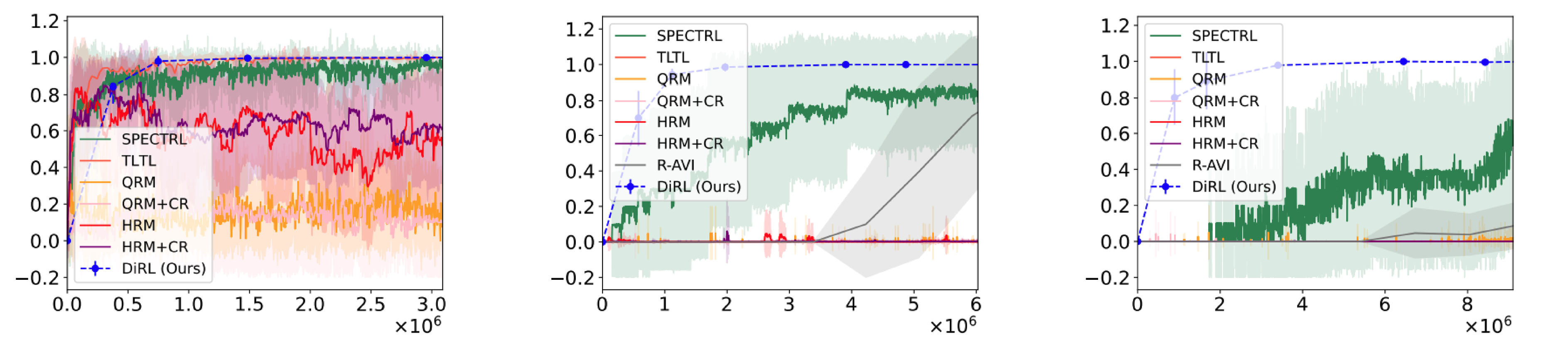
Phase II: Plan and Learn

Run ***Dijkstra's algorithm*** on the abstract graph

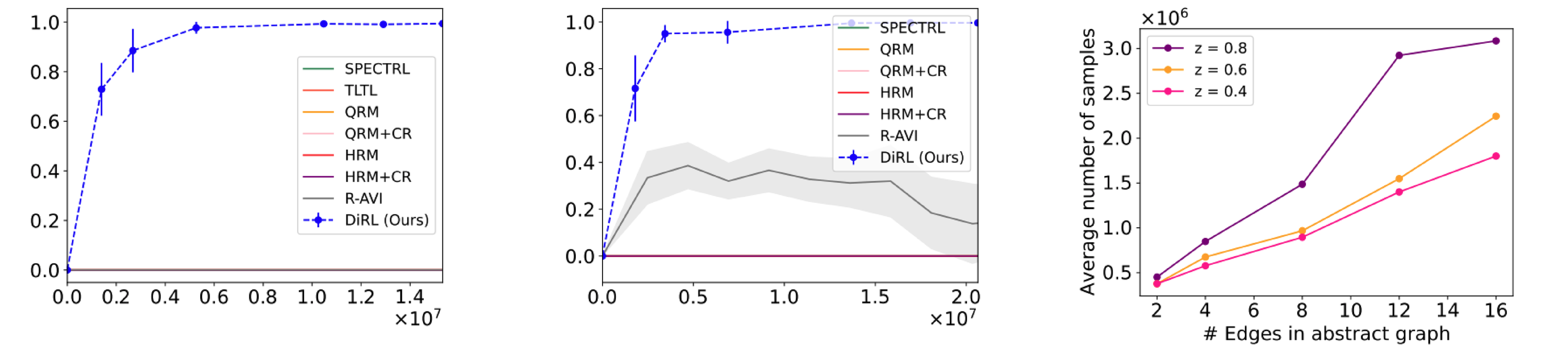
- Learn ***policy for an edge*** when Dijkstra's algorithm requires cost of edge
- Assign cost $-log(p_e)$ for edge e where p_e is the probability that subtask is completed successfully
- ***Initial state distribution*** chosen heuristically

Experiments

Rooms Environment

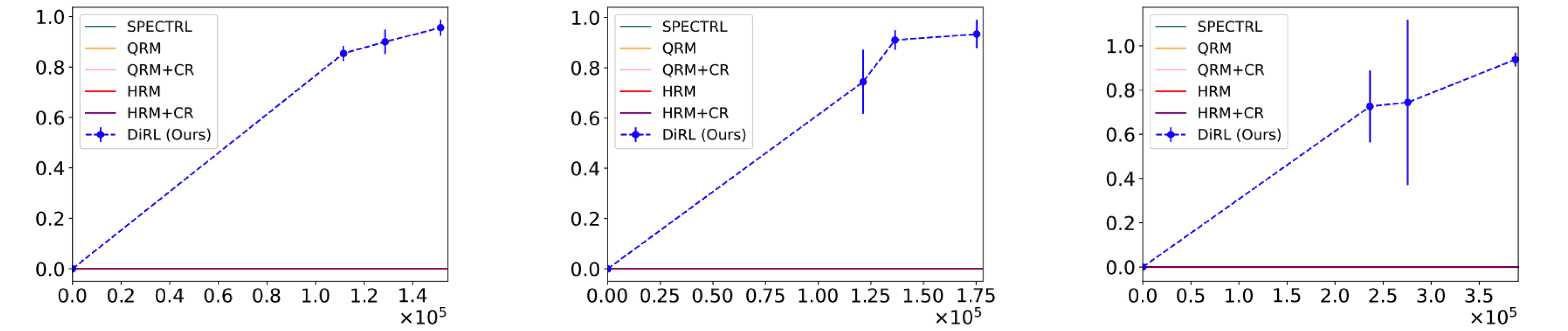


(a) Half sub-specification ϕ_1, $|\mathcal{G}_{\phi_1}| = 2$.
(b) 1 sub-specification ϕ_2, $|\mathcal{G}_{\phi_2}| = 4$.
(c) 2 sub-specifications ϕ_3, $|\mathcal{G}_{\phi_3}| = 8$.



(d) 3 sub-specifications ϕ_4, $|\mathcal{G}_{\phi_4}| = 12$.
(e) 4 sub-specifications ϕ_5, $|\mathcal{G}_{\phi_5}| = 16$.
(f) Sample complexity curves.

Fetch Environment



(a) PickAndPlace
(b) PickAndPlaceStatic
(c) PickAndPlaceChoice

Code

Paper