Symbolic Relational Deep RL based on Graph Neural Networks

Jaromír Janisch • **Tomáš Pevný** • **Viliam Lisý**

Artificial Intelligence Center / Department of Computer Science / FEE / Czech Technical University in Prague

Abstract

TL;DR: We present a generic Deep RL framework that can tackle any graph domain with objects and (multi-)parametrized actions and zero-shot generalizes to different domain sizes.

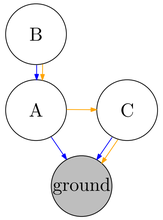
We focus on reinforcement learning (RL) in relational problems that are naturally defined in terms of objects, their relations, and manipulations. These problems are characterized by variable state and action spaces, and finding a fixed-length representation, required by most existing RL methods, is difficult, if not impossible. We present a deep RL framework based on graph neural networks and auto-regressive policy decomposition that naturally works with these problems and is completely domain-independent. We demonstrate the framework in three very distinct domains and we report the method's competitive performance and impressive zero-shot generalization over different problem sizes. In goal-oriented BlockWorld, we demonstrate multi-parameter actions with pre-conditions. In SysAdmin, we show how to select multiple objects simultaneously. In the classical planning domain of Sokoban, the method trained exclusively on 10×10 problems with three boxes solves 89% of 15×15 problems with five boxes.

Symbolic Representation with Graphs

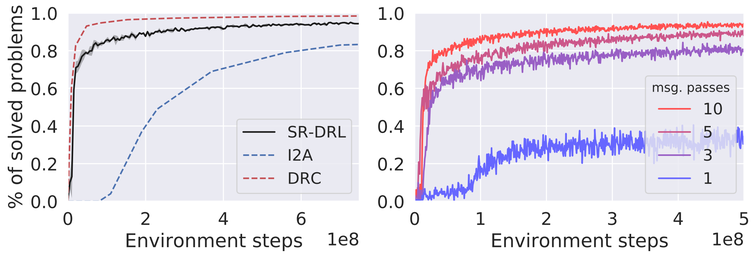
A state is represented as a heterogenous graph with **objects** as nodes and their **relations** as edges. A **goal** can be encoded in a global **context** or directly in the graph.

Actions are applied directly to the nodes, and can have **multiple parameters** – *move-box(from, to)*. **Pre-conditions** specify in which states an action can be applied.

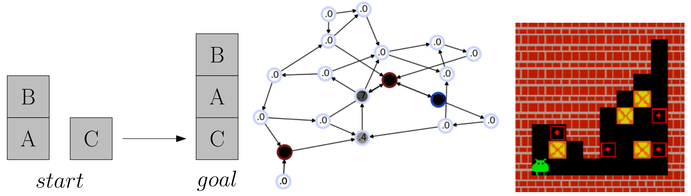
Set actions can be used with an arbitrary number of same-typed objects – *select(a, b, ...)*.



Training in Sokoban



Experiment Domains



(a) BlockWorld (b) SysAdmin (c) Sokoban

Method

GNN: The state is processed through a multi-layered GNN with a global node, skip connections and attention.

Actions: The action space is factored with auto-regressive policy decomposition;- $l = 0$ describes the action identifier (select, move-box, ...) and $l \geq 1$ parameters.

$$\pi(a|s) = \pi_0(a_0|s) \prod_{l=1}^{L(a)} \pi_{a_0,l}(a_l|a_{<l}, s)$$

Parameter selection: Selection of a parameter a_l is conditioned by previously selected $a_{<l}$. This is done by feeding the network back the $a_{<l}$ and performing a few message-passes again.

Set actions: Per-node probability $p(v)$ is computed and independently selected (Υ). A different decomposition is then used:

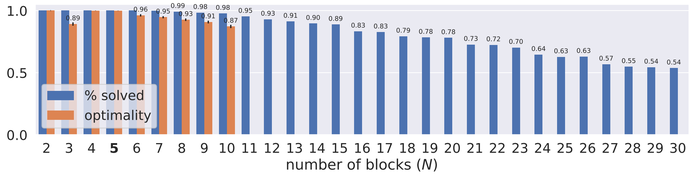
$$\pi(a|s) = \pi_0(a_0|g) \prod_{v \in \Upsilon} p(v) \prod_{v \in \mathcal{V} \setminus \Upsilon} (1 - p(v))$$

Training: A policy gradient algorithm (A2C) optimizes $\pi(a|s)$.

Acknowledgments: This research was supported by the European Office of Aerospace Research and Development (grant no. FA9550-18-1-7008) and by The Czech Science Foundation (grants no. 18-21409S and 18-27483Y). The GPU used for this research was donated by the NVIDIA Corporation. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures. The authors acknowledge the support of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

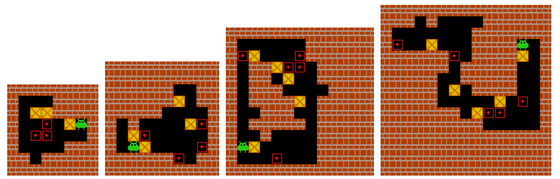
Experiment Results

BlockWorld



The agent trained only with **5 blocks** generalizes well to a different number.

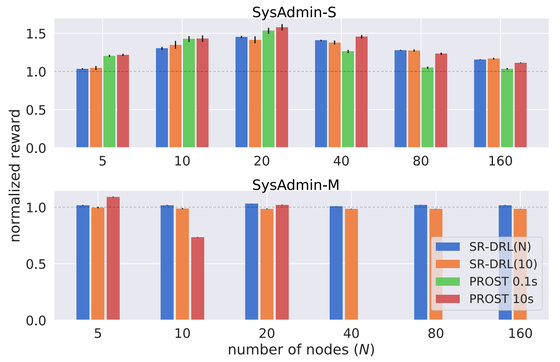
Sokoban



8×8 3 boxes	**10×10** **4 boxes**	13×13 5 boxes	15×15 5 boxes
96.7%	96%	86.9%	89%

The agent trained solely in **10×10 with 4 boxes** generalizes well to different sizes.

SysAdmin



The agent trained solely with **10 nodes** performs the same as if trained directly for the specified number. With 1 ms for a decision, it is competing with the PROST planner using much more time. With set actions, PROST cannot be used anymore for $N \geq 40$.