

How to Reduce Action Space for Planning Domains? (Student Abstract)

Harsha Kokel,^{1*} Junkyu Lee,² Michael Katz,² Shirin Sohrabi,² Kavitha Srinivas²

¹ The University of Texas at Dallas

² IBM Research

hkobel@utdallas.edu, {Junkyu.Lee,michael.katz1,kavitha.srinivas}@ibm.com, ssorab@us.ibm.com

Abstract

While AI planning and Reinforcement Learning (RL) solve sequential decision-making problems, they are based on different formalisms, which leads to a significant difference in their action spaces. When solving planning problems using RL algorithms, we have observed that a naive translation of the planning action space incurs severe degradation in sample complexity. In practice, those action spaces are often engineered manually in a domain-specific manner. In this abstract, we present a method that reduces the parameters of operators in AI planning domains by introducing a parameter seed set problem and casting it as a classical planning task. Our experiment shows that our proposed method significantly reduces the number of actions in the RL environments originating from AI planning domains.

Introduction

Recent success stories in Reinforcement Learning (RL) have renewed the interest in applying RL algorithms for solving AI planning tasks, as these tasks can be cast as a goal-oriented Markov decision process (MDP). The action space of an RL environment is then often obtained by a direct mapping from grounded PDDL (Planning Domain Definition Language) operators. The number of actions in the RL environment then rapidly grows as the number of objects in a planning task increases, which significantly degrades the performance of RL algorithms. To alleviate the issue, researchers have *manually* engineered the problem domains, either by generating the action space for the MDP independently from the PDDL action schema (Dzeroski, Raedt, and Driessens 2001), or by reducing the parameters of the PDDL action schema (Silver and Chitnis 2020). In this abstract, we define redundant parameters of action operators from the RL perspective and present an automated method of identifying these parameters using lifted mutex groups (Fiser 2020).

Preliminaries

In this section, we introduce notations for the normalized PDDL tasks and lifted mutex groups. A *normalized PDDL task* $\Pi := \langle \mathcal{L}, \mathcal{O}, I, G \rangle$ is defined over a first-order language

\mathcal{L} , a finite set of *schematic operators* \mathcal{O} , an *initial state specification* I , and a *goal specification* G . A schematic operator $o := \langle \text{head}(o), \text{pre}(o), \text{add}(o), \text{del}(o) \rangle$ consists of the atom $\text{head}(o)$ indicating the name of the operator, the preconditions $\text{pre}(o)$, the add effects $\text{add}(o)$, and the delete effects $\text{del}(o)$. We denote *parameters* of operator o by $\text{params}(o)$. An operator with an empty parameter set is called *ground operator*, and its head is a ground atom. We use notation $o_{\downarrow}(P/a)$ to denote a set of ground operators induced by assigning constants a to a subset of parameters P and grounding the remaining parameters with all possible constants. A *plan* for the normalized PDDL task Π is a sequence of ground operators leading the initial state to a goal state.

A *mutex group* M is a set of mutually exclusive ground predicates in any state s . For example, consider a gripper domain where a ball $b1$ can be placed in either room $r1$ or $r2$. Then, $\{at(b1, r1), at(b1, r2)\}$ is a mutex group since the ball $b1$ can only be in one of the rooms in any state. A *lifted mutex group* (LMG) is a lifted predicate that produces mutex groups when grounded. Formally, an LMG is a tuple $\langle v^f, v^c, \text{atom} \rangle$ with a finite set of fixed variables v^f , a finite set of counted variables v^c , and a finite set of atoms, atom . Let's consider an LMG $l := \langle \{?ball\}, \{?room\}, at(?ball, ?room) \rangle$ in the gripper domain where a robot with two grippers moves two balls $b1$ and $b2$ between two rooms $r1$ and $r2$. We can obtain two mutex groups: (1) $l_{\downarrow}(?ball/b1) = \{at(b1, r1), at(b1, r2)\}$ by assigning $b1$ to the fixed variable $?ball$ and grounding the count variable $?room$, and (2) $l_{\downarrow}(?ball/b2) = \{at(b2, r1), at(b2, r2)\}$ by assigning $b2$ to the fixed variable $?ball$. Note that different groundings of fixed variables $v^f(l)$ result in different sets of ground atoms, and the groundings of the counted variables $v^c(l)$ generate the ground atoms within the mutex group. We say a lifted mutex group l is **relevant** to the schematic operator o if $\text{atom}(l) \in \text{pre}(o)$, and Fiser (2020) provides a method for identifying a set of LMGs given a PDDL task Π .

Proposed Approach

The motivation of our work is to reduce the action space of an AI planning task, described as a goal-oriented MDP for RL. The set of RL actions L of such an MDP is composed of operator labels, one for each ground planning operator. We identify an assignment of labels to planning operators

*This work was partially done when HK was at IBM Research.
Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

such that it generates a smaller label set L' , while producing an equivalent transition system. We start by defining a *valid* label reduction. Next, we present a *parameter seed set problem*, and show the PDDL encoding for the problem.

Given two sets of labels L and L' , a label reduction from L to L' is **valid** if, for each reachable state, L' distinguishes any two outgoing transitions that are distinguished by L . From the planning perspective, at most one operator that corresponds to a reduced label may be applicable in a reachable state s . Formally, a set of operators $\mathcal{O}' \subseteq \mathcal{O}$ is an **applicable operator mutex group** (AOMG) if $|\{o \mid s \models \text{pre}(o), o \in \mathcal{O}'\}| \leq 1$ for any reachable state s . Naturally, a partitioning of operators into AOMGs defines a valid operator label reduction, and vice versa. Here, we find AOMGs separately for each lifted schematic operator, by removing some parameters from the schematic operator. For example, consider a lifted schematic operator $\text{pick}(\text{?ball}, \text{?room}, \text{?gripper})$ with three parameters $\{\text{?ball}, \text{?room}, \text{?gripper}\}$ in the gripper domain with two balls and two grippers. Since a gripper cannot be placed in different rooms in the same state, one possible set of AOMGs is a partition of the ground operators according to the assignments to the subset $\{\text{?ball}, \text{?gripper}\}$ of all parameters, $\{\{\text{pick}(b1, r1, g1), \text{pick}(b1, r2, g1)\}, \dots, \{\text{pick}(b2, r1, g2), \text{pick}(b2, r2, g2)\}\}$.

Given a lifted schematic operator o and an LMG $l = \langle v^f(l), v^c(l), \text{atom}(l) \rangle$, if $\text{atom}(l) \in \text{pre}(o)$, a set of ground operators $o_{\downarrow}(X/c)$ induced by assigning any constants c to $X = \text{params}(o) \setminus v^c(l)$ is an AOMG. If assignments to the fixed variables $v^f(l)$ are known, the assignments to the counted variables $v^c(l)$ can be uniquely identified in a state. Once these parameters are identified, their values are known, another LMG l' could be used to uniquely identify its own $v^c(l')$. Thus, we can iteratively reduce the subset of parameters that are required to be known. We formulate this iterative process as the following **parameter seed set** problem.

Input: A schematic operator o with parameters $\text{params}(o)$ and a set of *relevant* lifted mutex groups L .
Find: A subset $X \subseteq \text{params}(o)$ of parameters such that there exist X_1, \dots, X_k with (i) $X = X_1 \subseteq X_2 \subseteq \dots \subseteq X_k = \text{params}(o)$, and (ii) $X_{i+1} = X_i \cup v^c(l)$ for some $l \in L$ with $v^f(l) \subseteq X_i$.

Observe that any assignment to the solution of the parameter seed set problem corresponds to an AOMG.

To solve the parameter seed set problem, we encode it as a planning task $\Pi_o = \langle \mathcal{L}_o, \mathcal{O}_o, I_o, G_o \rangle$. The \mathcal{L}_o contains a single predicate mark and a constant symbol for each parameter in $\text{params}(o)$. The set of operators \mathcal{O}_o consists of two schematic operators seed and get_l , one per each relevant LMG l , where $\text{seed} := \langle \text{seed}(x), \emptyset, \{\text{mark}(x)\}, \emptyset \rangle$ and $\text{get}_l := \langle \text{get}_l, \{\text{mark}(x) \mid x \in v^f(l)\}, \{\text{mark}(y) \mid y \in v^c(l)\}, \emptyset \rangle$. The initial state I_o is an empty set \emptyset , and the goal is $G_o = \{\text{mark}(x) \mid \forall x \in \text{params}(o)\}$. Each plan π can be associated with $X_\pi = \{c \mid \text{seed}(c) \in \pi\}$, which is a set of constants extracted from the ground operators $\text{seed}(c)$ of the plan π . This subset X_π of $\text{params}(o)$ is a solution to the parameter seed set problem. For a schematic operator o , assigning any constants c to the seed-set X of its parameters

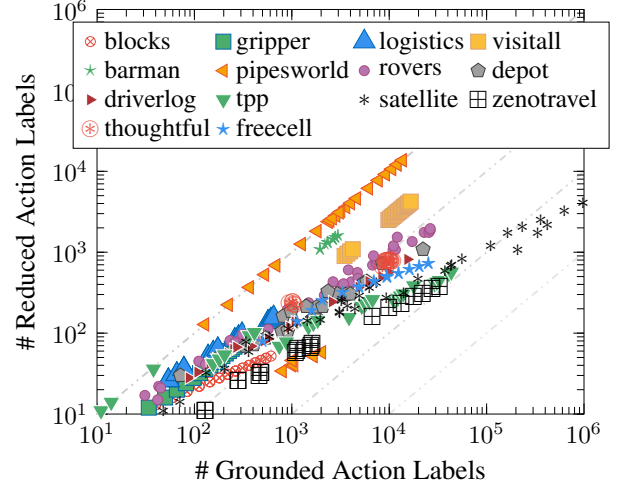


Figure 1: Reduction of Action Labels.

results in a set of ground operators $o_{\downarrow}(X/c)$ that is guaranteed to be an AOMG. Hence, all operators in that set can be assigned the same label when forming the MDP.

Evaluations and Conclusion

We implemented our approach using CPDDL for finding lifted mutex groups (Fiser 2020) and the Forbid-Iterative unordered top-quality planner (Katz, Sohrabi, and Udrea 2020) to find parameter seed-sets X . We evaluated fourteen AI planning domains from International Planning Competitions, and reported the reduction in the number of RL actions in Figure 1. For each problem instance, the x-axis shows the number of grounded operator labels before the reduction and the y-axis shows the number of reduced labels. Our approach shows a substantial reduction of the label set, going beyond 2 orders of magnitude on some instances. To evaluate the advantage of the action space reduction, we translated the PDDL task to a goal-oriented MDP with the reduced label set, and trained DDQN RL agents on four AI planning domains, *Blocks*, *Ferry*, *Gripper*, and *Logistics*. We observed that the reduction of action labels improved sample efficiency by approximately 300,000 steps in *Ferry* and *Gripper* domains, and more than 500,000 steps in *Blocks* and *Logistics*. Overall, our preliminary evaluation shows that the presented approach significantly reduced action spaces, resulting in improved sample efficiency of RL algorithms.

References

- Dzeroski, S.; Raedt, L. D.; and Driessens, K. 2001. Relational Reinforcement Learning. *Machine Learning*.
- Fiser, D. 2020. Lifted Fact-Alternating Mutex Groups and Pruned Grounding of Classical Planning Problems. In *AAAI*.
- Katz, M.; Sohrabi, S.; and Udrea, O. 2020. Top-Quality Planning: Finding Practically Useful Sets of Best Plans. In *AAAI*.
- Silver, T.; and Chitnis, R. 2020. PDDL Gym: Gym Environments from PDDL Problems. *CoRR*, abs/2002.06432.