Symmetries and Expressive Requirements for Learning General Policies

Dominik Drexler[1] Simon Ståhlberg[2] Blai Bonet[3] Hector Geffner[2]

November 7, 2024

[1] Linköping University, Linköping, Sweden,
[2] RWTH Aachen University, Aachen, Germany,
[3] Universitat Pompeu Fabra, Barcelona, Spain

Generalized Planning: Motivating Example

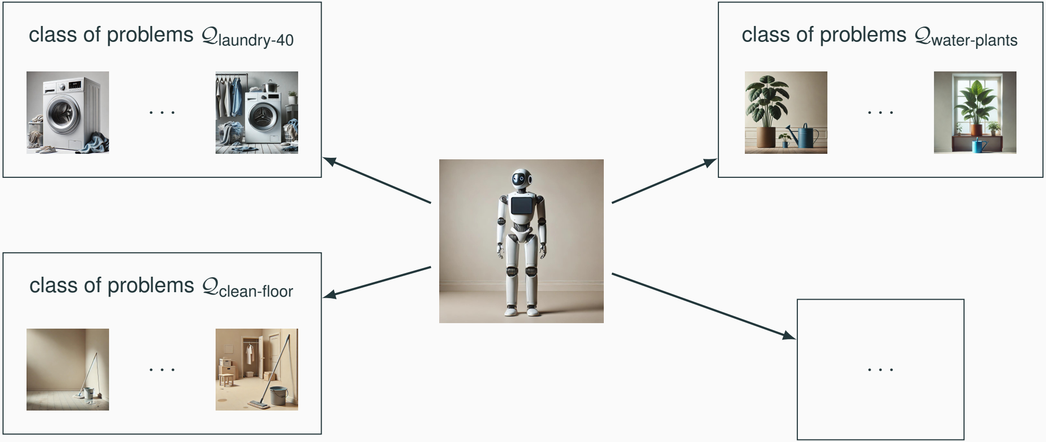
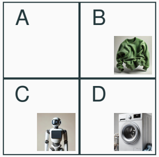


Figure 1: A robot facing several classes of problems. Images in this presentation created with Dall-E.

Formulation: Generalized Classical Planning

- **Given:** Class of problems $\mathcal{Q}$ consisting of **structurally similar** classical planning problems (over common first-order planning domain):
  - **Identical** relation symbols (to describe the world) and action schemas (to act in the world)
  - **Different** sets of objects, initial situation, or goal situation

(a) Problem 1 - initial situation: at(,A), at(,C), . . .



(b) Problem 2

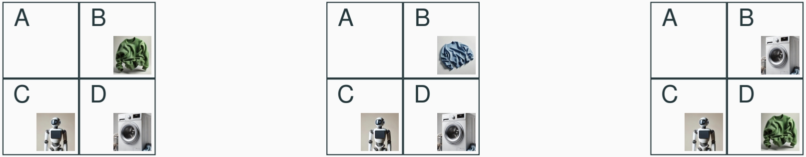
- Finding a plan for a single problem $P \in Q$ from scratch is **computationally challenging**
  $\rightarrow$ We want to **learn from experience** to make planning efficient
- **Objective:** Find a **general plan** $\mathcal{A}$ that efficiently (*polynomial time*) solves any problem $P \in \mathcal{Q}$
- **General policies** describe what action to take in a given situation (state) to reach goal in $P \in \mathcal{Q}$

General Policies

- A **general policy** π picks state transitions (s, s') in each $P \in \mathcal{Q}$
- A general policy π **solves** P if all π-trajectories starting at s_0 end in goal state
- A general policy π **solves** $\mathcal{Q}$ if it solves all P in $\mathcal{Q}$
- A general policy sees states through **features** over some feature language $\mathcal{L}$
- Feature language $\mathcal{L}$ **cannot** refer to objects by names
- E.g., feature as first-order logic sentence: $H \equiv \exists x.\text{holding}(x)$

In a Nutshell

- Others and us have been looking at two methods for **learning general policies**
  - **Combinatorial:** explicit pool of features, Weighted-Min-SAT formulation [Khardon, 1999, Martín and Geffner, 2004, Fern et al., 2006, Srivastava et al., 2008, Jiménez et al., 2019, Francès et al., 2021]
  - **Deep learning:** features learned to represent value or policy functions via DRL [Toyer et al., 2020, Bajpai et al., 2018, Rivlin et al., 2020, Ståhlberg et al., 2023]
- Two main issues:
  - **Scalability,** in combinatorial setting
  - **Expressivity,** in **both**
- Aims of this work:
  - Exploit **state symmetries** (isomorphisms) for reducing # of states in training
  - Use **symmetries** to evaluate the expressive requirements of planning domains



(a) State s_1 **(b)** State s_2: isomorphic to s_1 **(c)** State s_3: non-isomorphic

Figure 3: Two isomorphic states and one non-isomorphic state from problems of doing the laundry.

Equivalence-based Abstractions

- Planning states are **relational structures**
- Two states s, s' are **isomorphic** $s \sim_{iso} s'$ iff their relational structures are **isomorphic**
  $\rightarrow$ Isomorphism is a bijective relationship preserving mapping between objects from s to s'
  $\rightarrow$ Isomorphic states represent the **same problem aspect**
- **Reduced problems** $\tilde{\mathcal{Q}}$ contains one representative state from each class of isomorphic states
- **Theorem:** a general policy π solves $\mathcal{Q}$ iff π solves the **reduced problems** $\tilde{\mathcal{Q}}$.
  Detail: requires isomorphism-invariant feature language to work with representative state

How State Symmetries are Computed?

- States s mapped into **undirected vertex-colored graphs** $G(s)$
- Relation symbols can have arbitrary arity
- **Theorem:** $s \sim_{iso} s'$ iff $G(s) \sim_{iso} G(s')$
- We use state-of-the-art code (`nauty`) to determine if graphs $G(s)$ and $G(s')$ are **isomorphic**

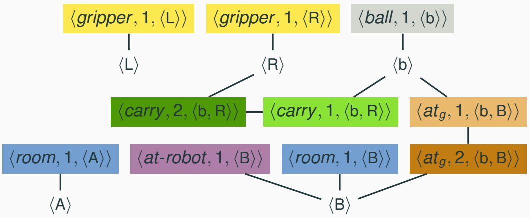


Figure 4: Graphs $G(s)$ for a state in a problem from the Gripper domain.

Experimental Results: Learning General Policies

Domain	Learning time (sec)	Learning speedup	Data reduction
Blocks3ops	11,233	**2.65**	**29.72**
Blocks4ops-clear	6	**1.33**	**355.12**
Blocks4ops-on	228	0.47	**122.65**
Delivery	325	**1.64**	**123.05**
Ferry	91	0.76	**31.81**
Gripper	6	0.83	**12.04**
Miconic	58	0.66	**2.63**
Reward	14	**1.43**	**1.91**
Spanner	8	0.88	**32.83**
Visitall	95	0.98	**1.18**

Table 1: Learning general policies with equivalence-based reductions.

Assessing Expressivity Requirements

- We can introduce a small twist to our pipeline to analyze the expressive requirements of feature languages on **training sets** from each planning domain
- Commonly used feature languages are description logics (combinatorial), GNNs (deep learning)
- Expressivity measured by the ability in distinguishing non-isomorphic states
- Failure to distinguish those states (**= conflict**) can result in failure to learn general policy
  $\rightarrow$ It might not be possible to assign different behaviors
- Straightforward method to assess expressivity requirements:
  - Run 1-WL on all representative (non-isomorphic) states of training set to find **conflicts**
  - **No conflict implies sufficient expressiveness** of C_2, GNNs, and description logics [Cai et al., 1992, Grohe, 2021]

Experimental Results: Expressivity Requirements

Domain	$\#\mathcal{Q}$	$\#\mathcal{S}$	$\#\mathcal{S}/{\sim_{iso}}$	# Conflicts 1-WL	# Conflicts 2-FWL
Barman	510	115 M	38 M	1,326	0
Blocks3ops	600	146 K	133 K	50	0
Blocks4ops	600	122 K	110 K	54	0
Blocks4ops-clear	120	31 K	3 K	0	0
Blocks4ops-on	150	31 K	8 K	0	0
Childsnack	30	58 K	5 K	0	0
Delivery	540	412 K	62 K	0	0
Ferry	180	8 K	4 K	36	0
Grid	1,799	438 K	370 K	42	0
Gripper	5	1 K	90	0	0
Hiking	720	44 M	5 M	0	0
Logistics	720	69 K	38 K	131	0
Miconic	360	32 K	22 K	0	0
Reward	240	14 K	11 K	0	0
Rovers	514	39 M	34 M	0	0
Satellite	960	14 M	8 M	5,304	0
Spanner	270	9 K	4 K	0	0
Visitall	660	3 M	2 M	0	0

Table 2: $\#\mathcal{Q}$ is # of problems; $\#\mathcal{S}$ and $\#\mathcal{S}/{\sim_{iso}}$: # states and partitions; # conflicts.

GNN + RL for General Policies [Ståhlberg et al., 2023]

		# Conflicts	
Domain	Coverage (%)	1-WL	2-FWL
Delivery	100%	0	0
Gripper	100%	0	0
Logistics	36%	131	0
Grid	79%	42	0

- Nearly **perfect general policies** obtained in several domains (100%)
- But interesting part is in the **failures**
  - **GNN expressivity not enough** (Logistics, Grid, Blocks)
  - Optimality-generality tradeoff
  - Others: insufficient # network layers, sampling
- More expressive GNN architectures look promising for obtaining general policies

Summary

- **Two methods** for learning general policies
  - **Combinatorial:** explicit pool of features, Weighted-Max-SAT formulation [Francès et al., 2021]
  - **GNNs:** features learned to represent value of policy functions via DRL [Ståhlberg et al., 2022]
- Two main issues:
  - **Scalability,** in combinatorial setting
  - **Expressivity,** in **both**
- **Computing symmetries**
  - Mapping states into graphs that preserve isomorphisms
  - Using state-of-the-art codes for testing graph isomorphism
- **Assessing expressivity**:
  - Helps in understanding failures
  - C_3 seems sufficient (= manageable upper-bound)

Bajpai, A. N., Garg, S., et al. (2018).
Transfer of deep reactive policies for mdp planning.
In *Proceedings of the Thirty-second Annual Conference on Neural Information Processing Systems (NeurIPS 2018)*, pages 10965–10975.

Cai, J.-Y., Fürer, M., and Immerman, N. (1992).
An optimal lower bound on the number of variables for graph identification.
Combinatorica, 12(4):389–410.

Fern, A., Yoon, S., and Givan, R. (2006).
Approximate policy iteration with a policy language bias: Solving relational markov decision processes.
Journal of Artificial Intelligence Research, 25:75–118.

Francès, G., Bonet, B., and Geffner, H. (2021).
Learning general planning policies from small examples without supervision.
In Leyton-Brown, K. and Mausam, editors, *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021)*, pages 11801–11808. AAAI Press.

Grohe, M. (2021).

The logic of graph neural networks.
In *Proceedings of the Thirty-Sixth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2021)*, pages 1–17.

Jiménez, S., Segovia-Aguas, J., and Jonsson, A. (2019).
A review of generalized planning.
The Knowledge Engineering Review, 34:e5.

Khardon, R. (1999).
Learning action strategies for planning domains.
Artificial Intelligence, 113:125–148.

Martín, M. and Geffner, H. (2004).
Learning generalized policies from planning examples using concept languages.
Applied Intelligence, 20(1):9–19.

Rivlin, O., Hazan, T., and Karpas, E. (2020).
Generalized planning with deep reinforcement learning.
In *ICAPS 2020 Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning (PRL)*, pages 16–24.

Srivastava, S., Immerman, N., and Zilberstein, S. (2008).

Learning generalized plans using abstract counting.
In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 991–997. AAAI Press.

Ståhlberg, S., Bonet, B., and Geffner, H. (2022).
Learning generalized policies without supervision using GNNs.
In *Proceedings of the Nineteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2022)*, pages 474–483.

Ståhlberg, S., Bonet, B., and Geffner, H. (2023).
Learning general policies with policy gradient methods.
In *Proceedings of the Twentieth International Conference on Principles of Knowledge Representation and Reasoning (KR 2023)*, pages 647–657.

Toyer, S., Thiébaux, S., Trevizan, F., and Xie, L. (2020).
ASNets: Deep learning for generalised planning.
Journal of Artificial Intelligence Research, 68:1–68.