Learning General Policies and Helpful Action Classifiers from Partial State Spaces



Dominik Drexler[1] Javier Segovia-Aguas[2] Jendrik Seipp[1]

July 23, 2022

[1]Linköping University, Linköping, Sweden,
[2]Universitat Pompeu Fabra, Barcelona, Spain

Motivation

- A **general policy** encodes domain general strategies
  $\rightarrow$ Only exist for tractable domains
- How to learn **partial policies** for **intractable domains**?
- How to learn general policies for **difficult but tractable domains**?
- Idea: find policy that solves **partial state spaces**

Example General Policy for Visitall

Features Φ

- d: distance to nearest unvisited cell
- n: number of unvisited cells

Policy rules π_Φ	Meaning
$\{d > 0, n > 0\} \mapsto \{d\downarrow\}$	; decrease distance to nearest unvisited cell
$\{d > 0, n > 0\} \mapsto \{d?, n\downarrow\}$	; visit unvisited cell

Syntax and Semantics of Policies [Francès et al., 2021]

,

- **Syntax:**
  - **Policy** π_Φ consists of **policy rules** of form $C \mapsto E$ over features Φ
  - For **Boolean feature** p and **numerical feature** n, we can have
    - $p, \neg p, n > 0, n = 0$ in C
    - $p, \neg p, p?, n\uparrow, n\downarrow, n?$ in E
- **Semantics:**
  - **Transition** (s, a, s') is π_Φ-**compatible** iff
    1. $s \models C$, and
    2. $(s, s') \models E$
  - A policy is **general** if for every alive state there exists π_Φ-compatible transition
  - A policy is **partial** if there exists an alive state with no π_Φ-compatible transition
  - In both cases, **acyclicity** over the π_Φ-compatible transitions is required

Learning Policies from Complete State Spaces

The learning problem [Francès et al., 2021]

Given: **fully expanded** state spaces $S(P_1), \ldots, S(P_n)$, and feature pool $\mathcal{F}$

Find: **policy** π_Φ with $\Phi \subseteq \mathcal{F}$, s.t., π_Φ solves all $S(P_1), \ldots, S(P_n)$

- Generality of policy across whole domain can often be proven
- No provable generalization capabilities in intractable domains
- W-Max-SAT encoding: scalability problems, even more in intractable domains

$\rightarrow$ **Idea:** observe only fragments of the state space

Learning Policies from Partial State Spaces

The learning problem

Given: **partially expanded** state spaces $\mathcal{S}(P_1), \ldots, \mathcal{S}(P_n)$, and feature pool $\mathcal{F}$

Find: **policy** π_Φ with $\Phi \subseteq \mathcal{F}$, s.t., π_Φ solves all $\mathcal{S}(P_1), \ldots, \mathcal{S}(P_n)$

- How to obtain an informative fragment of the state space?
  $\rightarrow$ **Idea:** optimal plans contain goal directed information

Learning Policies from Partial State Spaces: Details

- Notation: expanded states $\mathcal{S}$, generated states $\mathcal{G}$
- Initially, $\mathcal{G}_0$ only contains initial states s_0, $\mathcal{S} = \emptyset$, $\pi_\Phi = \emptyset$
- In each iteration $i = 1, 2, \ldots$
  - Sample one **optimal** *s*-**plan** p for $s \in \mathcal{G}_{i-1}$ where π_Φ fails
  - $\mathcal{S}_i = \mathcal{S}_{i-1} \ \cup \ \{$ alive states on p $\}$
  - $\mathcal{G}_i = \mathcal{G}_{i-1} \ \cup \ \{$ alive states on p $\} \cup \{$ 1-step successor states s along p $\}$
  - Generate feature pool $\mathcal{F}_i$ with respect to $\mathcal{G}_i$
  - Solve W-Max-SAT encoding $\Gamma(\mathcal{S}_i, \mathcal{G}_i, \mathcal{F}_i)$ to find π_Φ that solves $\mathcal{S}_i$ **suboptimally**
  - Aim for "simplest" policy: minimize $\sum_{f \in \Phi} complexity(f)$

Helpful Actions with Partial Policies

- Need additional **thinking** if policy is partial
- Combine heuristic search with policy: GBFS with h_{FF} and helpful actions [Hoffmann and Nebel, 2001]
- Dual-queue approach: one prioritized queue for states reached by helpful actions
- **Action** a is π_Φ-**helpful** in state s if transition (s, a, s') is π_Φ-compatible
- **Action** a is **relaxed-helpful** in state s if a is applicable in s and part of a relaxed plan from s

Experiments: Configurations

- All configurations use GBFS and h_{FF}
  - FF: –
  - FF^r: dual-queue and **relaxed-helpful** actions
  - FF^π: dual-queue and π_Φ-**helpful** actions
  - FF_∞: like FF but **greedily** following the policy after each expansion step
  - FF^r_∞: like FF^r but **greedily** following the policy after each expansion step
- Search properties: suboptimal, sound and complete

Experiments

	FF			FF^r			FF^π			FF_∞			FF^r_∞		
	S	E	T	S	E	T	S	E	T	S	E	T	S	E	T
Barman (30)	6	0.05	0.08	**23**	**0.50**	**0.34**	8	0.06	0.09	7	0.08	0.11	20	0.39	0.28
Blocks (30)	26	0.85	0.48	26	0.86	0.48	26	0.86	0.47	24	0.80	0.44	26	0.86	0.43
Blocks-clear (30)	**30**	0.98	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	0.99	**30**	**1.00**	0.99
Blocks-on (30)	**30**	0.95	**1.00**	**30**	**1.00**	**1.00**	**30**	0.95	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**
Childsnack (30)	1	0.00	0.01	7	0.12	0.16	3	0.02	0.03	1	0.02	0.03	**8**	**0.20**	**0.21**
Delivery (30)	**30**	0.99	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**
Depots (30)	5	0.15	0.20	**6**	**0.18**	**0.21**	5	0.10	0.12	**6**	0.16	0.18	5	0.16	0.17
Driverlog (30)	8	0.13	0.16	**18**	**0.34**	**0.29**	4	0.06	0.07	6	0.10	0.10	7	0.14	0.13
Ferry (30)	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**
Freecell (30)	27	0.49	0.63	26	**0.54**	**0.65**	26	0.40	0.57	26	0.48	0.62	**27**	0.52	**0.65**
Gripper (30)	**30**	0.61	0.81	**30**	0.68	0.80	**30**	0.54	0.57	**30**	**0.87**	0.94	**30**	**0.87**	0.94
Miconic (30)	**30**	**0.90**	0.97	**30**	**0.90**	**0.96**	**30**	**0.90**	0.76	**30**	**0.90**	0.84	**30**	**0.90**	0.84
N-puzzle (30)	**30**	0.88	**1.00**	**30**	0.87	**1.00**	**30**	**0.89**	**1.00**	**30**	0.88	**1.00**	**30**	0.87	**1.00**
Nomystery (30)	7	0.09	0.13	**10**	**0.22**	**0.26**	3	0.05	0.02	6	0.17	0.11	6	0.16	0.11
Parking (30)	9	0.21	0.18	11	0.29	0.25	6	0.10	0.08	**18**	**0.51**	**0.35**	17	0.49	**0.35**
Pipes-nt (30)	12	0.17	0.26	**24**	0.52	0.62	13	0.24	0.32	11	0.15	0.22	**24**	**0.63**	**0.71**
Pipes-t (30)	11	0.14	0.22	**28**	**0.65**	**0.63**	13	0.17	0.24	12	0.16	0.23	25	0.61	0.61
Reward (30)	**30**	0.92	**1.00**	**30**	0.94	**1.00**	**30**	**0.99**	0.99	**30**	**0.99**	**1.00**	**30**	**0.99**	**1.00**
Satellite (30)	10	0.33	0.36	**14**	**0.47**	**0.44**	10	0.31	0.34	10	0.33	0.36	13	0.43	0.42
Sokoban (30)	19	0.24	0.45	**22**	**0.26**	**0.46**	15	0.24	0.32	14	0.24	0.32	13	0.23	0.31
Spanner (30)	0	0.00	0.00	0	0.00	0.00	27	0.74	0.13	**30**	**0.87**	0.30	**30**	**0.87**	0.30
Visitall (30)	5	0.05	0.11	5	0.06	0.11	6	0.12	0.14	**25**	**0.59**	**0.44**	**25**	**0.59**	**0.44**
Zenotravel (30)	12	0.34	0.43	**15**	**0.49**	**0.67**	9	0.21	0.26	9	0.24	0.35	**15**	0.45	0.55
Sum (690)	398	0.45	0.50	475	0.56	**0.58**	414	0.48	0.46	445	0.54	0.52	**501**	**0.62**	**0.58**

Experiments

	FF			FF^r			FF^π			FF_∞			FF^r_∞		
	S	E	T	S	E	T	S	E	T	S	E	T	S	E	T
Barman (30)	6	0.05	0.08	**23**	**0.50**	**0.34**	8	0.06	0.09	7	0.08	0.11	20	0.39	0.28
Blocks (30)	26	0.85	0.48	26	0.86	0.48	26	0.86	0.47	24	0.80	0.44	26	0.86	0.43
Blocks-clear (30)	**30**	0.98	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	0.99	**30**	**1.00**	0.99
Blocks-on (30)	**30**	0.95	**1.00**	**30**	**1.00**	**1.00**	**30**	0.95	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**
Childsnack (30)	1	0.00	0.01	7	0.12	0.16	3	0.02	0.03	1	0.02	0.03	**8**	**0.20**	**0.21**
Delivery (30)	**30**	0.99	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**
Depots (30)	5	0.15	0.20	**6**	**0.18**	**0.21**	5	0.10	0.12	**6**	0.16	0.18	5	0.16	0.17
Driverlog (30)	8	0.13	0.16	**18**	**0.34**	**0.29**	4	0.06	0.07	6	0.10	0.10	7	0.14	0.13
Ferry (30)	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**	**30**	**1.00**	**1.00**
Freecell (30)	27	0.49	0.63	26	**0.54**	**0.65**	26	0.40	0.57	26	0.48	0.62	**27**	0.52	**0.65**
Gripper (30)	**30**	0.61	0.81	**30**	0.68	0.80	**30**	0.54	0.57	**30**	**0.87**	0.94	**30**	**0.87**	0.94
Miconic (30)	**30**	**0.90**	0.97	**30**	**0.90**	**0.96**	**30**	**0.90**	0.76	**30**	**0.90**	0.84	**30**	**0.90**	0.84
N-puzzle (30)	**30**	0.88	**1.00**	**30**	0.87	**1.00**	**30**	**0.89**	**1.00**	**30**	0.88	**1.00**	**30**	0.87	**1.00**
Nomystery (30)	7	0.09	0.13	**10**	**0.22**	**0.26**	3	0.05	0.02	6	0.17	0.11	6	0.16	0.11
Parking (30)	9	0.21	0.18	11	0.29	0.25	6	0.10	0.08	**18**	**0.51**	**0.35**	17	0.49	**0.35**
Pipes-nt (30)	12	0.17	0.26	**24**	0.52	0.62	13	0.24	0.32	11	0.15	0.22	**24**	**0.63**	**0.71**
Pipes-t (30)	11	0.14	0.22	**28**	**0.65**	**0.63**	13	0.17	0.24	12	0.16	0.23	25	0.61	0.61
Reward (30)	**30**	0.92	**1.00**	**30**	0.94	**1.00**	**30**	**0.99**	0.99	**30**	**0.99**	**1.00**	**30**	**0.99**	**1.00**
Satellite (30)	10	0.33	0.36	**14**	**0.47**	**0.44**	10	0.31	0.34	10	0.33	0.36	13	0.43	0.42
Sokoban (30)	19	0.24	0.45	**22**	**0.26**	**0.46**	15	0.24	0.32	14	0.24	0.32	13	0.23	0.31
Spanner (30)	0	0.00	0.00	0	0.00	0.00	27	0.74	0.13	**30**	**0.87**	0.30	**30**	**0.87**	0.30
Visitall (30)	5	0.05	0.11	5	0.06	0.11	6	0.12	0.14	**25**	**0.59**	**0.44**	**25**	**0.59**	**0.44**
Zenotravel (30)	12	0.34	0.43	**15**	**0.49**	**0.67**	9	0.21	0.26	9	0.24	0.35	**15**	0.45	0.55
Sum (690)	398	0.45	0.50	475	0.56	**0.58**	414	0.48	0.46	445	0.54	0.52	**501**	**0.62**	**0.58**

Future Work & Conclusions

- We presented a method for combining heuristic search with partial policies learned from partial state spaces
- How to obtain more informed fragments of the state space?
- How to identify tractable fragments?

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.

Hoffmann, J. and Nebel, B. (2001).
The FF planning system: Fast plan generation through heuristic search.
Journal of Artificial Intelligence Research, 14:253–302.