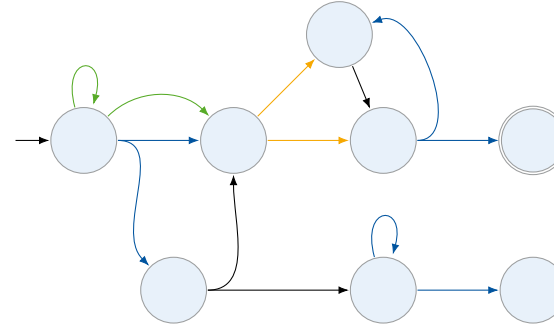
Learning Generalized Policies for Fully Observable Non-Deterministic Planning Domains

IJCAI 2024



Till Hofmann, Hector Geffner

August 6, 2024

Chair of Computer Science 6 (Machine Learning and Reasoning)

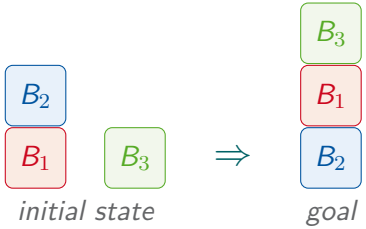
RWTH AACHEN UNIVERSITY

Generalized Policies for Classical Planning

Classical planning:

- Given: action model + initial state + goal
- Task: find a (concrete) policy $\pi_P : S \rightarrow 2^A$ that maps each state s to a set of actions $\pi_P(s)$

```
(:action pickup :parameters (?x)
 :precondition (and
   (clear ?x) (ontable ?x) (handempty))
 :effect (and (not (ontable ?x))
   (not (clear ?x)) (not (handempty))
   (holding ?x)))
(:action putdown ...)
(:action stack ...)
(:action unstack ...)
```



Generalized Policies for Classical Planning

Classical planning:

- Given: action model + initial state + goal
- Task: find a (concrete) policy $\pi_P : S \to 2^A$ that maps each state s to a set of actions $\pi_P(s)$

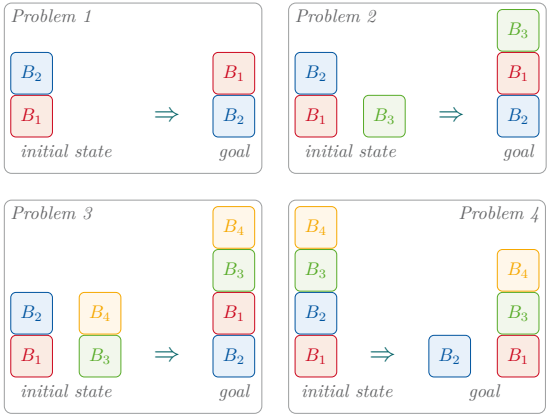
Generalized planning:

- Given: action model + **class of problems** $\mathcal{Q}$
- *Each* problem: initial state + goal
- Task: find a general policy that maps each problem $P \in \mathcal{Q}$ to a concrete policy $\pi(P) = \pi_P$

Example (Blocks World)

General policy:

1. Unstack all misplaced blocks
2. Stack them in the correct order



Generalized Policies for Classical Planning

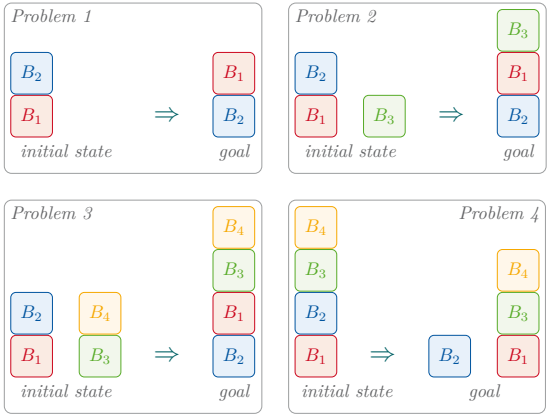
Classical planning:

- Given: action model + initial state + goal
- Task: find a (concrete) policy $\pi_P : S \to 2^A$ that maps each state s to a set of actions $\pi_P(s)$

Generalized planning:

assumed to be *closed*

- Given: action model + **class of problems** $\mathcal{Q}$
- *Each* problem: initial state + goal
- Task: find a general policy that maps each problem $P \in \mathcal{Q}$ to a concrete policy $\pi(P) = \pi_P$



Example (Blocks World)

General policy:

1. Unstack all misplaced blocks
2. Stack them in the correct order

Expressing and Learning Generalized Policies

Rule-based general policies (Bonet and Geffner 2018):

- Express a policy as a set of rules of the form $C \mapsto E_1 \mid \ldots \mid E_k$ over a feature space Φ
- Features: Boolean and numerical features, e.g., *# clear blocks*
- Learnable from a set of small problems (Bonet et al. 2019; Francès et al. 2021)

Example

General policy for clearing a one block x:

$$r_1 : \{\neg H, n > 0\} \mapsto \{H, n\downarrow\}$$
$$r_2 : \quad \{H\} \mapsto \{\neg H\}$$

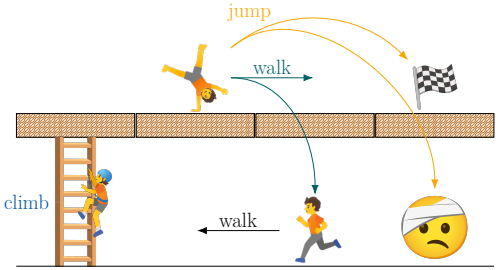
where

$H \triangleq$ true if holding a block

$n \triangleq$ number of blocks on top of x

Generalized Policies for FOND Planning

- FOND: action a may have multiple outcomes $a_1, a_2, \ldots$ not under the agent's control
- *Fairness:* for an infinitely occurring action a, all outcomes $a_1, a_2, \ldots$ occur infinitely often
- *Strong-cyclic:* all max. fair trajectories reach goal

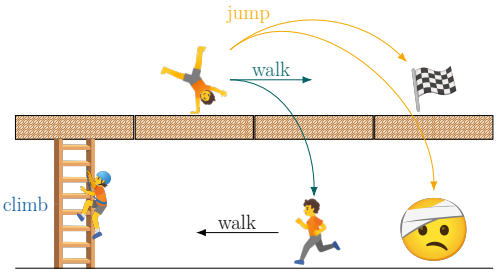


Generalized Policies for FOND Planning

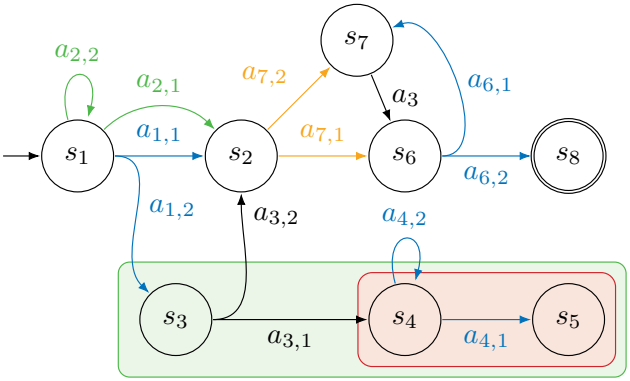
- FOND: action a may have multiple outcomes $a_1, a_2, \ldots$ not under the agent's control
- *Fairness:* for an infinitely occurring action a, all outcomes $a_1, a_2, \ldots$ occur infinitely often
- *Strong-cyclic:* all max. fair trajectories reach goal

Questions

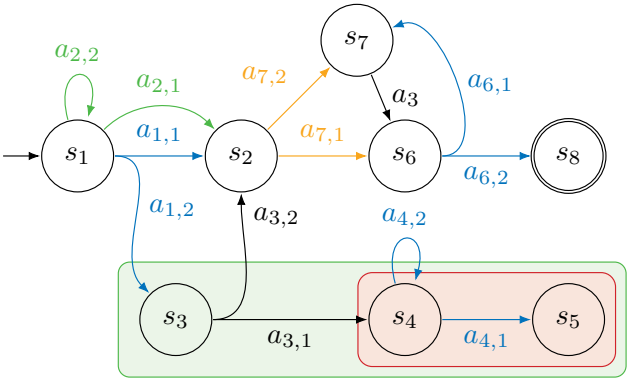
1. How do we express general policies for FOND domains?
2. How can we learn FOND general policies?



From Classical to FOND General Policies

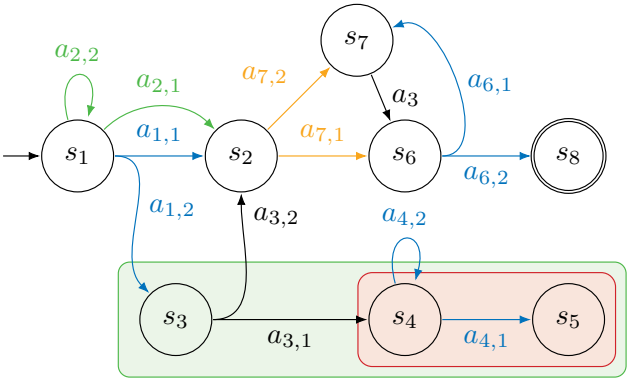


From Classical to FOND General Policies



All-outcome relaxation: agent can choose action outcome → classical planning

From Classical to FOND General Policies

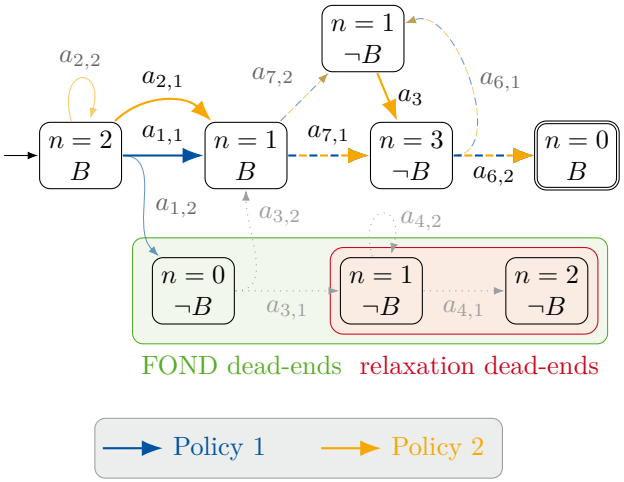


All-outcome relaxation: agent can choose action outcome $\rightarrow$ classical planning

Observation

A general policy for the *all-outcome relaxation* is also a general policy for the original FOND problem if it avoids FOND dead-ends

From Classical to FOND General Policies

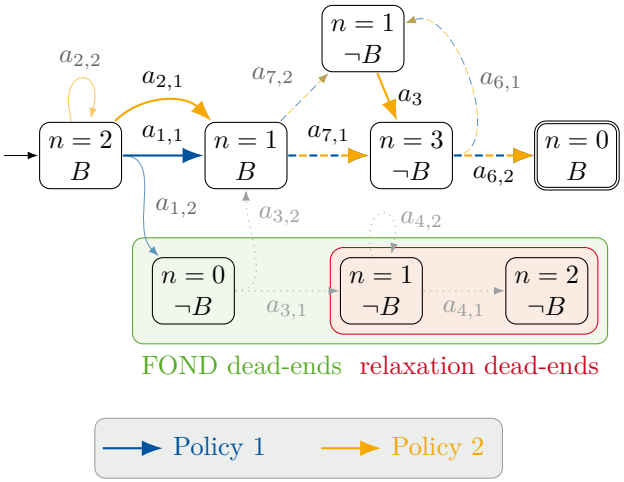


All-outcome relaxation: agent can choose action outcome → classical planning

Observation

A general policy for the *all-outcome relaxation* is also a general policy for the original FOND problem if it avoids FOND dead-ends

From Classical to FOND General Policies



All-outcome relaxation: agent can choose action outcome → classical planning

Observation

A general policy for the *all-outcome relaxation* is also a general policy for the original FOND problem if it avoids FOND dead-ends

Learning:

- Select features that describe the FOND dead-ends
- Learn a general policy for the relaxation that avoids dead-ends

⇒ Combinatorial optimization problem

more details on poster

Example: Learned Policy for Acrobatics

Selected features:

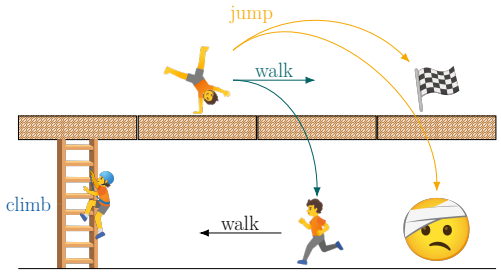
1. Distance $d \equiv \mathrm{dist}(\mathit{position}, \mathit{next\text{-}fwd}, \mathit{position}_G)$ between the current position and the goal position
2. Boolean feature $U \equiv |\mathit{up}|$ which is true if the agent is currently on the beam
3. Boolean feature $B \equiv |\mathit{broken\text{-}leg}|$ which is true if the agent's leg is broken

Learned policy rules:

$$r_1 : \quad \{U, d > 0, \neg B\} \mapsto \{d\downarrow\}$$
$$r_2 : \quad \{\neg B, \neg U\} \mapsto \{U\} \mid \{d\uparrow\}$$

Learned state constraint:

$$b_1 : \quad \{B, \neg U\}$$



Evaluation

$\mathcal{Q}$	# problems	# solved	# training p.	# obj (train)	# obj (all)	t_{solve}/s	# features	# sel. features	# constraints	max complex.	policy cost
acrobatics	18	18	3	3	9	0.12	23	3	1	4	6
beam-walk	9	9	2	3	9	0.05	22	2	0	4	5
blocks3ops	95	95	4	4	20	3:44	194	3	0	5	11
blocks-clear	95	95	2	3	20	1.5	34	2	0	4	6
blocks-on	190	190	2	3	20	1:56	704	3	0	6	11
doors	19	19	5	7	33	1:18	625	4	1	10	19
islands	300	300	4	32	83	1h	1182	4	1	7	13
first-resp	99	**15**$^{\mathsf{M}}$	2	5	36	33:40	332	5	2	7	20
miner	69	**13**$^{\mathsf{I}}$	2	9	184	297h	1073	8	4	6	28
spiky-tire	170	**36**$^{\mathsf{I}}$	3	6	23	1h	479	8	5	8	36
tireworld	980	**7**$^{\mathsf{C}}$	1	3	100	0.11	27	5	4	4	12
triangle-tire	10	**1**$^{\mathsf{I}}$	1	6	231	0.29	27	3	1	4	9

Evaluation

$\mathcal{Q}$	# problems	# solved	# training p.	# obj (train)	# obj (all)	t_{solve}/s	# features	# sel. features	# constraints	max complex.	policy cost
acrobatics	18	18	3	3	9	0.12	23	3	1	4	6
beam-walk	9	9	2	3	9	0.05	22	2	0	4	5
blocks3ops	95	95	4	4	20	3:44	194	3	0	5	11
blocks-clear	95	95	2	3	20	1.5	34	2	0	4	6
blocks-on	190	190	2	3	20	1:56	704	3	0	6	11
doors	19	19	5	7	33	1:18	625	4	1	10	19
islands	300	300	4	32	83	1h	1182	4	1	7	13
first-resp	99	**15**$^{\mathsf{M}}$	2	5	36	33:40	332	5	2	7	20
miner	69	**13**$^{\mathsf{I}}$	2	9	184	297h	1073	8	4	6	28
spiky-tire	170	**36**$^{\mathsf{I}}$	3	6	23	1h	479	8	5	8	36
tireworld	980	**7**$^{\mathsf{C}}$	1	3	100	0.11	27	5	4	4	12
triangle-tire	10	**1**$^{\mathsf{I}}$	1	6	231	0.29	27	3	1	4	9

Evaluation

$\mathcal{Q}$	# problems	# solved	# training p.	# obj (train)	# obj (all)	t_{solve}/s	# features	# sel. features	# constraints	max complex.	policy cost
acrobatics	18	18	3	3	9	0.12	23	3	1	4	6
beam-walk	9	9	2	3	9	0.05	22	2	0	4	5
blocks3ops	95	95	4	4	20	3:44	194	3	0	5	11
blocks-clear	95	95	2	3	20	1.5	34	2	0	4	6
blocks-on	190	190	2	3	20	1:56	704	3	0	6	11
doors	19	19	5	7	33	1:18	625	4	1	10	19
islands	300	300	4	32	83	1h	1182	4	1	7	13
first-resp	99	**15**$^{\mathsf{M}}$	2	5	36	33:40	332	5	2	7	20
miner	69	**13**$^{\mathsf{I}}$	2	9	184	297h	1073	8	4	6	28
spiky-tire	170	**36**$^{\mathsf{I}}$	3	6	23	1h	479	8	5	8	36
tireworld	980	**7**$^{\mathsf{C}}$	1	3	100	0.11	27	5	4	4	12
triangle-tire	10	**1**$^{\mathsf{I}}$	1	6	231	0.29	27	3	1	4	9

Evaluation

$\mathcal{Q}$	# problems	# solved	# training p.	# obj (train)	# obj (all)	t_{solve}/s	# features	# sel. features	# constraints	max complex.	policy cost
acrobatics	18	18	3	3	9	0.12	23	3	1	4	6
beam-walk	9	9	2	3	9	0.05	22	2	0	4	5
blocks3ops	95	95	4	4	20	3:44	194	3	0	5	11
blocks-clear	95	95	2	3	20	1.5	34	2	0	4	6
blocks-on	190	190	2	3	20	1:56	704	3	0	6	11
doors	19	19	5	7	33	1:18	625	4	1	10	19
islands	300	300	4	32	83	1h	1182	4	1	7	13
first-resp	99	**15**$^{\mathsf{M}}$	2	5	36	33:40	332	5	2	7	20
miner	69	**13**$^{\mathsf{I}}$	2	9	184	297h	1073	8	4	6	28
spiky-tire	170	**36**$^{\mathsf{I}}$	3	6	23	1h	479	8	5	8	36
tireworld	980	**7**$^{\mathsf{C}}$	1	3	100	0.11	27	5	4	4	12
triangle-tire	10	**1**$^{\mathsf{I}}$	1	6	231	0.29	27	3	1	4	9

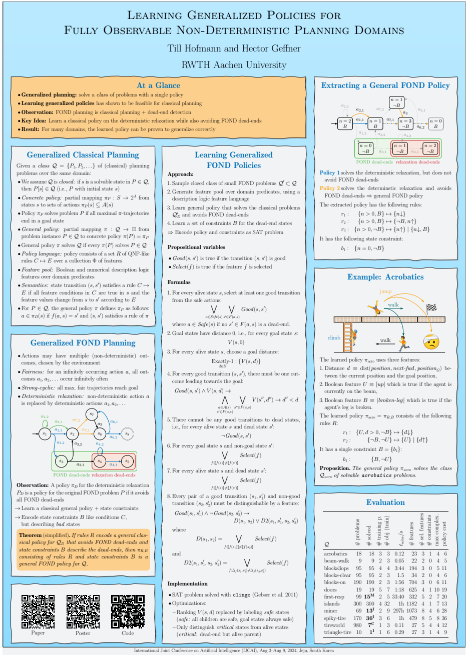
Evaluation

$\mathcal{Q}$	# problems	# solved	# training p.	# obj (train)	# obj (all)	t_{solve}/s	# features	# sel. features	# constraints	max complex.	policy cost
acrobatics	18	18	3	3	9	0.12	23	3	1	4	6
beam-walk	9	9	2	3	9	0.05	22	2	0	4	5
blocks3ops	95	95	4	4	20	3:44	194	3	0	5	11
blocks-clear	95	95	2	3	20	1.5	34	2	0	4	6
blocks-on	190	190	2	3	20	1:56	704	3	0	6	11
doors	19	19	5	7	33	1:18	625	4	1	10	19
islands	300	300	4	32	83	1h	1182	4	1	7	13
first-resp	99	**15**$^{\mathsf{M}}$	2	5	36	33:40	332	5	2	7	20
miner	69	**13**$^{\mathsf{I}}$	2	9	184	297h	1073	8	4	6	28
spiky-tire	170	**36**$^{\mathsf{I}}$	3	6	23	1h	479	8	5	8	36
tireworld	980	**7**$^{\mathsf{C}}$	1	3	100	0.11	27	5	4	4	12
triangle-tire	10	**1**$^{\mathsf{I}}$	1	6	231	0.29	27	3	1	4	9

Evaluation

proven to generalize to all instances

$\mathcal{Q}$	# problems	# solved	# training p.	# obj (train)	# obj (all)	t_{solve}/s	# features	# sel. features	# constraints	max complex.	policy cost
acrobatics	18	18	3	3	9	0.12	23	3	1	4	6
beam-walk	9	9	2	3	9	0.05	22	2	0	4	5
blocks3ops	95	95	4	4	20	3:44	194	3	0	5	11
blocks-clear	95	95	2	3	20	1.5	34	2	0	4	6
blocks-on	190	190	2	3	20	1:56	704	3	0	6	11
doors	19	19	5	7	33	1:18	625	4	1	10	19
islands	300	300	4	32	83	1h	1182	4	1	7	13
first-resp	99	**15**$^{\mathsf{M}}$	2	5	36	33:40	332	5	2	7	20
miner	69	**13**$^{\mathsf{I}}$	2	9	184	297h	1073	8	4	6	28
spiky-tire	170	**36**$^{\mathsf{I}}$	3	6	23	1h	479	8	5	8	36
tireworld	980	**7**$^{\mathsf{C}}$	1	3	100	0.11	27	5	4	4	12
triangle-tire	10	**1**$^{\mathsf{I}}$	1	6	231	0.29	27	3	1	4	9



Come visit our poster!

Paper

Appendix

Bibliography (I)

- Bonet, Blai, Guillem Francès, and Hector Geffner (2019). "Learning features and abstract actions for computing generalized plans". In: *AAAI*, pp. 2703–2710.
- Bonet, Blai and Hector Geffner (2018). "Features, Projections, and Representation Change for Generalized Planning". In: *IJCAI*, pp. 4667–4673.
- Francès, Guillem, Blai Bonet, and Hector Geffner (2021). "Learning general planning policies from small examples without supervision". In: *AAAI*, pp. 11801–11808.

Till Hofmann, Hector Geffner

IJCAI, Jeju Island, South Korea, August 6, 2024