Learning Sketches for Decomposing Planning Problems into Subproblems of Bounded Width

Dominik Drexler[1]

Jendrik Seipp[1]

Hector Geffner[2,1]

June 21, 2022

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

Motivation

- Two important questions in planning (and RL) are:
  1. What is a good language for representing the subgoal structure of planning tasks?
     $\rightarrow$ Policy sketches [Bonet and Geffner, 2021]
  2. How to learn common subgoal structure of a family of tasks?
     $\rightarrow$ In this paper

Policy Sketches

- Policy sketches (sketches) are simple and powerful [Drexler et al., 2021]
- Sketch splits problems into subproblems of bounded width in such a way that problems become solvable in polynomial time by the SIW_R algorithm
- Semantics in terms of what subgoal to achieve
- Not so much: more complex languages such as HTN or LTL

Outline

- Example
- Sketches
- Learning sketches of width k
- Experimental results

Example Sketch for the Delivery Domain

Features Φ

- H: holding a package?
- n: number of undelivered packages
- p: distance to nearest package
- t: distance to target cell

Rules R_Φ

$\{n > 0\} \mapsto \{n\downarrow\}$; deliver misplaced package

Example Sketch for the Delivery Domain (cont.)

Rules R_Φ; 2-width sketch

$\{n > 0\} \mapsto \{n\downarrow\}$; deliver misplaced package

Rules R_Φ; 1-width sketch

$\{\neg H\} \mapsto \{H\}$; pick pkg

$\{H, n > 0\} \mapsto \{H?, n\downarrow\}$; deliver pkg

Rules R_Φ; 0-width sketch or general policy [Francès et al., 2021]

$\{\neg H, p > 0\} \mapsto \{p\downarrow, t?\}$; go to nearest pkg

$\{\neg H, p = 0\} \mapsto \{H\}$; pick it up

$\{H, t > 0\} \mapsto \{t\downarrow\}$; go to target

$\{H, n > 0, t = 0\} \mapsto \{H?, n\downarrow, p?\}$; deliver pkg

Syntax and Semantics of Sketches

- **Syntax:**
  - **Sketch rule** has form $C \mapsto E$
  - 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:**
  - State pair (s, s') **satisfies** sketch rule $C \mapsto E$ if
    1. s satisfies C, and
    2. (s, s') satisfied E

Sketch Width

- Sketch R splits problem P in $\mathcal{Q}$ into collection of subproblems $P[s, G_R(s)]$ where
  - initial state s is reachable state s in P, and
  - (sub) goal states $G_R(s) = \{s' \mid (s, s')$ satisfies sketch rule or s' is goal$\}$
- **Width of problem** $\mathrm{w}(P[s, G])$ is exploitable measure for difficulty of achieving goal G from initial state s [Lipovetzky and Geffner, 2012]
- **Width of sketch** R over $\mathcal{Q}$ is $\max\{w(P[s, G_R(s)]) \mid s \in P, P \in \mathcal{Q}\}$
- **Theorem**: Any P in $\mathcal{Q}$ solvable with $\exp(k)$ resources if sketch has width k and sketch is terminating

Example Sketch for the Floortile Domain

Features Φ

- n: number of painted tiles
- S: state is solvable?

Rules R_Φ

$\{S, n > 0\} \mapsto \{n\downarrow\}$; deliver misplaced package

Theorem

The sketch R_Φ for the Floortile domain is terminating and has width 2.

Learning Sketches as Combinatorial Optimization

- **Given:**
  - Planning tasks $P_1, \ldots, P_n$
  - Feature pool $\mathcal{F}$
  - Sketch width k
  - Maximum number of rules m
- **Find:** sketch R_Φ over features $\Phi \subseteq \mathcal{F}$ with m rules that
  1. results in subproblems $P[s, G_R(s)]$ of width $\leq k$,
  2. is acyclic in each P_i (approximation of termination), and
  3. has minimum feature complexity, i.e., $\sum_{f \in \Phi} \text{complexity}(f)$

Learning Sketches as Combinatorial Optimization: Details

- Select R_Φ consisting of m rules
  - **Construct rules:** $cond(i, f, v)$, $eff(i, f, v)$, use unique v, implies $select(f)$
  - **Ensure compatibility:** $sat\_rule(s, s', i)$ iff (s, s') compatible with rule i
- Ensure that R_Φ is terminating
  - **Ensure termination:** collection of rules $i = 1, \ldots, m$ is terminating
- Ensure that R_Φ has sketch width $\leq k$
  - **Select subgoal tuples:** $\vee_t subgoal(s, t)$, each alive s has some subgoal t
  - **Select subgoal states:** $subgoal(s, t)$ iff $\wedge_{s'} subgoals(s, t, s')$
  - **Ensure compatible rule:** $subgoals(s, t, s')$ **implies** $\vee_{i=1,m} sat\_rule(s, s', i)$
  - **Ensure deadend free:** $sat\_rule(s, s'', i)$ implies $\vee_{t:d(s,t)<d(s,s'')} subgoal(s, t)$
  - **Ensure optimal width:** $sat\_rule(s, s', i)$ implies $\vee_{t:d(s,t)\leq d(s,s')} subgoal(s, t)$
- Implementation as answer set program in Clingo [Gebser et al., 2012]

Experimental Results of Learning Sketches

Table 1: Learning results for width bound $k = 1$, maximum feature complexity of 8, time limit of 7 days, and memory limit of 384 GiB.

Domain	Memory	Time	$\lvert\mathcal{P}\rvert$	$\lvert$States$\rvert$	$\lvert\mathcal{F}\rvert$	max. feature complexity	$\lvert\Phi\rvert$	$\lvert R\rvert$
Blocks-clear	1	4	1	22	233	4	1	1
Blocks-on	9	105	1	22	1011	4	2	2
Childsnack	122	228k	3	792	629	6	4	5
Delivery	17	521	1	96	474	4	2	2
Gripper	3	60	1	28	301	4	2	2
Miconic	1	5	1	32	119	2	2	2
Reward	1	4	1	12	210	2	1	1
Spanner	3	22	1	74	424	5	1	1
Visitall	1	1	1	3	10	2	1	1

Experimental Results of Testing the Learned Sketches

Table 2: Testing results for time limit 30 minutes and 6 GiB memory.

	$w = 0$		$w = 1$		$w = 2$		LAMA		BFWS	
Domain	Solved	Time	S	T	S	T	S	T	S	T
Blocks-clear (30)	30	3	30	5	30	4	30	4	30	6
Blocks-on (30)	30	3	30	6	30	3	30	4	30	25
Childsnack (30)	–	–	30	1	–	–	9	2	5	658
Delivery (30)	–	–	30	1	30	4	30	1	30	1
Gripper (30)	30	4	30	3	30	656	30	1	30	6
Miconic (30)	–	–	30	5	30	132	30	7	30	25
Reward (30)	30	4	30	2	30	1	30	2	30	1
Spanner (30)	30	3	30	4	30	3	0	–	0	–
Visitall (30)	26	1360	30	20	30	21	29	213	25	833
#Domains solved (9)	5		**9**		8		6		6	

Formal Properties of Learned Sketches

- Sketch width $\leq k$ only guaranteed for training instances $P_1, \ldots, P_n$
- However, sketch width $\leq k$ across family of tasks $\mathcal{Q}$ was proven

Learned Sketch for the Visitall Domain

Features Φ

- n: number visited locations

Rules R_Φ

$\{\} \mapsto \{n\!\uparrow\}$; visit a new location

Theorem

The sketch R_Φ for the Visitall domain is acyclic and has width 1.

Learned Sketch for the Childsnack Domain

Features Φ

- *sk*: number of *sandwiches* at the *kitchen*,
- *ua*: number of *unserved* and *allergic* children,
- *gfs*: number of *gluten-free sandwiches*, and
- *s*: number of *served* children.

Rules R_Φ

$\{\} \mapsto \{gfs\uparrow\}$	; make gluten free sandwiches
$\{\} \mapsto \{sk\downarrow\}$	; move sandwiches from kitchen on tray
$\{\} \mapsto \{ua\downarrow\}$	; serve gluten-free sandwich to allergic children
$\{ua = 0\} \mapsto \{sk\uparrow\}$	; make any sandwich **if** all allergic children are served
$\{ua = 0\} \mapsto \{s\uparrow\}$	; serve arbitrary sandwhich **if** all allergic children are served

Theorem

The sketch R_Φ for the Childsnack domain is acyclic and has width 1.

Limitations

- Sketches with bounded width ensure poly time solutions and hence only possible for tractable domains
- Learning implementation in Clingo does not scale up in all domains, e.g., Barman, Schedule, Floortile, Driverlog
- Feature pool assumes first-order language to describe states (PDDL)

Conclusions and Future Work

- First general method for learning how to decompose planning problems into subproblems with a polynomial complexity that is controlled with a parameter
- Future work:
  - From sketches to hierarchies
  - From PDDL inputs/states to other state languages

Bonet, B. and Geffner, H. (2021).
General policies, representations, and planning width.
In Leyton-Brown, K. and Mausam, editors, *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021)*, pages 11764–11773. AAAI Press.

Drexler, D., Seipp, J., and Geffner, H. (2021).
Expressing and exploiting the common subgoal structure of classical planning domains using sketches.
In Erdem, E., Bienvenu, M., and Lakemeyer, G., editors, *Proceedings of the Eighteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2021)*, pages 258–268.

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.

Gebser, M., Kaminski, R., Kaufmann, B., and Schaub, T. (2012).
Answer Set Solving in Practice.
Morgan & Claypool Publishers.

Lipovetzky, N. and Geffner, H. (2012).
Width and serialization of classical planning problems.
In De Raedt, L., Bessiere, C., Dubois, D., Doherty, P., Frasconi, P., Heintz, F., and Lucas, P., editors, *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, pages 540–545. IOS Press.