Planning to Avoid Side Effects

Toryn Q. Klassen*, Sheila A. McIlraith*, Christian Muise†, Jarvis Xu†

* Department of Computer Science, University of Toronto
Vector Institute for Artificial Intelligence
Schwartz Reisman Institute for Technology and Society
† School of Computing, Queen's University

Introduction to Side Effects in AI Safety

Underspecified objectives may lead an AI system to cause negative **side effects** (Amodei et al., 2016).

- A robot directed to go to a location may **break a vase** on the shortest path (Amodei et al.).

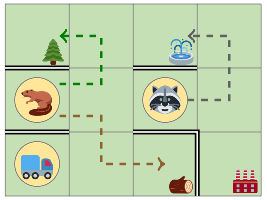
There are various works on avoiding or learning to avoid side effects in **MDPs** (e.g., Turner, Hadfield-Menell, and Tadepalli, 2020; Krakovna et al., 2020; Saisubramanian, Kamar, and Zilberstein, 2020).

Are Side Effects a Risk for Classical Planning?

- Symbolic planning problems were often **designed by hand** and didn't offer much opportunity for negative side effects.
- **Problem-specific** symbols may not even be able to represent side effects.
- But more **realistically complicated** or **learned** models may present risks that can be avoided.

Contributions

- **formalize** the notion of side effect in classical planning
- define classes of **negative side effects** relating to **impact on other agents' ability** to subsequently realize their goals and plans
- provide mechanisms for **computing** side-effect-minimizing plans for STRIPS problems



Canadian Wildlife domain

Background: Symbolic Planning and STRIPS

A **state-transition system** is a tuple $\langle S, A, \delta \rangle$ where

- S is a finite set of states
- A is a finite set of actions
- $\delta : S \times A \to S$ is a partial function

A **planning problem** consists of

- a state transition system $\langle S, A, \delta \rangle$
- an initial state $s_0 \in S$
- a set of goal states $S_G \subseteq S$

A **plan** is an action sequence $\pi = a_1, a_2, \ldots, a_k$ reaching a goal state.

In **STRIPS** planning problems:

- a set of **fluents** are used to represent properties that can change, e.g., `at_robot_A` could represent whether a robot is at location A
- a **state** is represented by a set of fluents (those true in that state)
- the **goal** is a set of fluents which have to be made true (while the other fluents can take any value), e.g., {`at_robot_B`}

Abstract Version of Minimizing Side Effects

Given a planning problem and **distance function** $d : S \times S \to [0, \infty)$, a plan π is **change-minimizing** if it minimizes the distance between the initial and final states (see also the discussion of distance functions by Amodei et al.).

All of the types of side effect minimization we'll consider can be thought of as special cases of this.

Fluent Side Effects

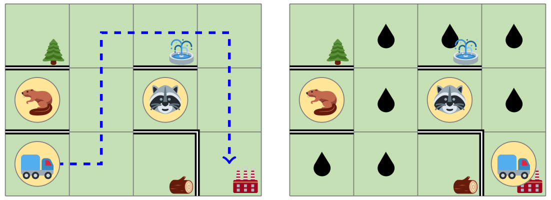
A fluent f is a side effect of a plan π if f is true after executing π, even though f was neither initially true nor part of the goal. Similarly, $\neg f$ is a side effect if f was initially true.

Fluent-Preserving Plan

A plan π for a STRIPS planning problem is **fluent-preserving** if no other plan has strictly fewer fluent side effects.

Goal Side Effects

- Given a multi-agent planning environment, suppose that **agent** i can achieve a **goal** $\hat{S}_G$ from the initial state.
- A plan π has a **goal side effect on agent** i w.r.t. goal $\hat{S}_G$ if i **can no longer achieve** $\hat{S}_G$ after π is executed.

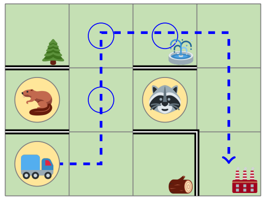


The truck going to the factory leaves a trail of oil, blocking the animals.

Goal-Preserving Plan

Given a planning problem, a set H of goal-agent pairs (s.t. the given agent initially can achieve the goal), and a weight function $w : H \to \mathbb{R}$, a plan π is **goal-preserving** if it **minimizes the weighted sum of goals** from H that are **made unachievable for their corresponding agents.**

- Suppose H consists of reaching , reaching , and reaching .
- The plan in which cleans the circled cells allows to reach , and to reach .
- That is a **goal-preserving plan** to reach the factory if only 3 cells can be cleaned, and the goals are equally weighted.



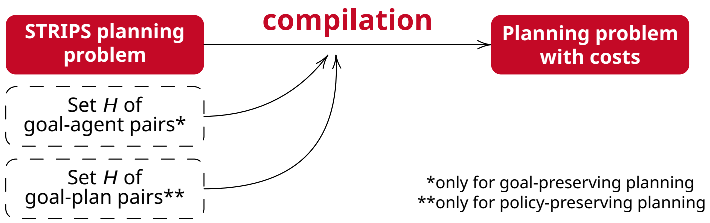
Policy Side Effects

- A (partial) policy is a (partial) function from states to actions.
- Given a multi-agent planning environment, suppose that **agent** i can achieve a goal $\hat{S}_G$ from the initial state **using policy** ρ.
- A plan π has a **policy side effect on agent** i w.r.t. goal $\hat{S}_G$ and policy ρ if i **can no longer achieve** $\hat{S}_G$ **using** ρ after π is executed.

Policy-Preserving Plan

Given a planning problem, a set H of goal-**policy** pairs (s.t. the given policy initially can achieve the goal), and a weight function $w : H \to \mathbb{R}$, a plan π is **policy-preserving** if it **minimizes the weighted sum of goals** from H **made unachievable by their corresponding policies.**

Computation



Compilation Details

The approach is based on the **soft goals** compilation by Keyder and Geffner (2009).

fluent-preserving: each fluent true in the initial state, and negation of a fluent that's false in the initial state, is made a soft goal

policy-preserving: the policies are represented using plans, and **regression** is used to determine the conditions that would have to hold for them to reach their goals

goal-preserving: the agent tries to find a plan in which as many goals as possible from H are achieved in sequence by their corresponding agents, with the environment being reset in between

Experimental Results

|H|: number of goal-policy / goal-agent pairs
PT: planning time (seconds)
CT: compilation time (seconds)
FSE: fluent side effects
PSE: policy side effects
GSE: goal side effects

Domain & Problem	$\lvert H \rvert$	Standard planning				Fluent-preserving					Policy-preserving			Goal-preserving		
		FSE	PSE	GSE	PT	FSE	PSE	GSE	CT	PT	PSE	CT	PT	GSE	CT	PT
wildlife	3, 3	17	3	3	0.5	13	3	3	0.8	20.2	1	0.6	6.5	1	0.6	38.0
zeno-a	5, 2	7	4	0	0.5	5	4	0	17.6	10.6	3	17.6	9.5	0	17.3	23.3
zeno-b	4, 2	5	2	0	0.4	5	2	0	17.6	7.2	0	17.4	10.4	0	17.0	24.6
zeno-c	7, 4	5	3	0	0.4	3	3	0	18.2	12.3	3	17.9	7.9	0	17.2	26.3
floortile-a	4, 2	6	4	0	0.5	2	3	1	2.8	16.9	0	2.5	9.2	0	2.5	56.4
floortile-b	4, 2	5	4	0	0.4	1	3	0	2.8	11.6	0	2.4	7.3	0	2.5	54.6
floortile-c	8, 4	5	8	1	0.5	1	5	0	2.8	18.5	1	2.5	4.9	0	2.5	97.2
storage-a	6, 2	5	5	0	0.4	5	5	0	0.9	7.4	0	0.9	10.4	0	0.9	14.1
storage-b	4, 2	8	4	0	0.4	5	2	0	0.9	6.2	0	0.9	5.2	0	0.9	15.5
storage-c	7, 4	14	3	2	0.4	10	3	0	0.9	7.0	3	0.9	5.7	0	0.9	16.2
storage-c2	7, 4	14	3	2	0.4	10	3	0	10.2	44.0	3	10.0	48.8	0	10.1	21.0
storage-c3	7, 4	14	3	2	0.4	10	3	0	49.8	163.5	3	50.3	159.3	0	48.5	53.7

Future Work

- side effects before the plan's end
- side effects on others' plan costs
- trade-off between plan cost and side effects
- more efficient ways of minimizing side effects

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané (2016). "Concrete Problems in AI Safety". In: *arXiv preprint arXiv:1606.06565*.

Emil Keyder and Hector Geffner (2009). "Soft Goals Can Be Compiled Away". In: *JAIR* 36, pp. 547–556. DOI: 10.1613/jair.2857.

Victoria Krakovna, Laurent Orseau, Richard Ngo, Miljan Martic, and Shane Legg (2020). "Avoiding Side Effects By Considering Future Tasks". In: *NeurIPS 2020*.

Sandhya Saisubramanian, Ece Kamar, and Shlomo Zilberstein (2020). "A Multi-Objective Approach to Mitigate Negative Side Effects". In: *IJCAI 2020*, pp. 354–361. DOI: 10.24963/ijcai.2020/50.

Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli (2020). "Conservative Agency via Attainable Utility Preservation". In: *AIES '20*, pp. 385–391. DOI: 10.1145/3375627.3375851.