Planning to Avoid Side Effects (Preliminary Report)

Toryn Q. Klassen[1] and **Sheila A. McIlraith**[1,2]
[1]University of Toronto and Vector Institute
[2]Schwartz Reisman Institute for Technology and Society
{toryn, sheila}@cs.toronto.edu

Abstract

In sequential decision making, objective specifications are often underspecified or incomplete, neglecting to take into account potential (negative) side effects. In this paper we investigate how to avoid side effects in a classical planning setting. We formalize several versions of side-effect-minimizing objectives for use in classical planning. We also study the notion of minimizing side effects in the context of a planning environment where multiple independent agents co-exist in a shared environment. We consider how side effects may compromise or prevent agents in the shared environment from achieving goals or executing plans that they would have otherwise been able to achieve, had the acting agent not executed their plan, and propose means to generate plans that minimize such negative side effects. Finally, we show how side-effect-minimizing plans can be computed via STRIPS planning with soft goals.

1 Introduction

Sequential decision making – planning – relies on the specification of an objective; typically a final-state goal condition in the case of symbolic planning, or a reward function in the case of reinforcement learning. Such objectives tend to be underspecified or incomplete, enabling the AI, pursuant of its objective, to realize additional (often discretionary) changes to the environment, which we refer to as *side effects*. In some cases these side effects are of little consequence, while in other cases they may change the environment in ways that are highly undesirable. Amodei *et al.* [2016] give the example of a robot breaking a vase that it wasn't explicitly told not to break, noting that "for an agent operating in a large, multifaceted environment, an objective function that focuses on only one aspect of the environment may implicitly express indifference over other aspects of the environment". A number of approaches for avoiding or learning to avoid negative side effects have recently been developed for Markov Decision Processes (MDPs) and related formalisms [Zhang *et al.*, 2018; Krakovna *et al.*, 2019; Turner *et al.*, 2020; Krakovna *et al.*, 2020; Saisubramanian *et al.*, 2020a]. Here, we explore how to avoid side effects in the context of symbolic planning. Awareness and avoidance of undesirable side effects is central to robust and reliable symbolic planning, in particular, and sequential decision making more generally.

The underspecification of objectives is but one of a series of representation and reasoning challenges that have faced the reasoning about action and planning communities over the years. The *frame problem* sought to deal with the challenge of parsimoniously encoding the non-effects of actions within a transition system model in a manner that would render the intended interpretation [McCarthy and Hayes, 1969]. The *qualification problem*, perhaps still unsolved satisfactorily, sought to deal with the challenge with enumerating all the preconditions of an action [McCarthy, 1977]. McCarthy's well-known quip on this topic related to the car not starting if there is a potato in the tailpipe. Common to addressing many of these challenges was determining how to fill in what was not stated in a manner that was both systematic and commonsensical. Common to the solutions was a notion of minimization – minimizing change in the case of the frame problem, where techniques such as circumscription were used to minimize change in certain fluents while allowing others to vary. We see avoiding side effects as one aspect of an *objective interpretation problem* that, at times, appeals to minimization.

The impact of the underspecification of planning objectives and the threat of negative side effects has always existed, but has been amplified as planning is increasingly applied or transferred to open world environments where the complete effects of actions cannot always be foreseen, immediately observed, or codified. Interestingly, whether a side effect is construed as negative or positive is often determined by those affected by the change. In the absence of definitive information, a conservative stance is for an acting agent planning to achieve a goal to generate a plan that minimizes all side effects – to leave the world in a state as close to the way it was prior to the execution of its plan.

In a multiagent setting, side effects can be assessed in the context of their impact on the agency of other agents. Here, we consider how one agent's actions may prevent other agents from achieving goals or executing plans that they would have otherwise been able to achieve, and propose means to generate plans that minimize such negative side effects (given knowledge about possible goals or plans of other agents).

The main contributions of this paper are the following:

1. We characterize the notion of side effect in classical planning.
2. We formalize several versions of side-effect-minimizing objectives for use in classical planning. These characterizations are realized via a notion of minimizing change in general, or change specific to particular properties.
3. We characterize classes of negative side effects that relate to the impact of an acting agent's plan on other agents' ability to subsequently realize their goals and plans. This is in contrast to some recent work in side effects that takes into account only how the agent's actions will affect its own future abilities [Krakovna *et al.*, 2019; Turner *et al.*, 2020; Krakovna *et al.*, 2020].
4. We propose mechanisms for computing side-effect-minimizing plans in STRIPS planning problems by compiling the task of achieving that objective into a STRIPS problem with *soft goals* [Keyder and Geffner, 2009], which can then be solved using established techniques.

2 Preliminaries

In this section we review definitions and notation related to classical planning.

Notation relating to sets. Given a set A, we will use the notation $|A|$ to denote the cardinality of A, and A^* to denote the set of all finite sequences of elements from A. We sometimes abbreviate a sequence $a_1, \ldots, a_k$ as $\vec{a}$.

We define a planning problem in terms of a state transition system following Ghallab *et al.* [2004, p. 17] and Ghallab *et al.* [2016, Definition 2.1].

Definition 1 (State-transition system)**.** 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, and $\delta : S \times A \to S$ is a partial function.

Notation. We extend the definition of δ to take a sequence of actions as an argument. If $\delta(s_0, a_1) = s_1, \delta(s_1, a_2) = s_2, \ldots, \delta(s_{k-1}, a_k) = s_k$, then $\delta(s_0, a_1, \ldots, a_k) = s_k$.

Definition 2 (Planning problem)**.** A *planning problem* is a tuple $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ where $\Sigma = \langle S, A, \delta \rangle$ is a state-transition system, $s_0 \in S$ is the initial state, and $S_G \subseteq S$ is the set of goal states.

Definition 3 (Plan for $\mathcal{P}$)**.** Given a planning problem $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ and a sequence of actions $\pi = a_1, a_2, \ldots, a_k$, $\pi \in A^*$, π is a plan for $\mathcal{P}$ iff $\delta(s_0, a_1, \ldots, a_k) \in S_G$.

In the above definitions we make no commitment to the nature of the states in our planning problem. They could be comprised of pixels or propositions. In symbolic planning, a state is typically defined in terms of a set of propositions that establish the truth or falsity of properties of the state. In classical planning a state s is represented compactly in terms of the set of propositions (*fluents*) that are true in the state and all fluents not in the set are regarded as false, like a database. The transition function is similarly represented compactly in terms of a set of database operations that add or delete propositions from the database when an action is performed. The truth or falsity of all other propositions persists (the frame assumption). This class of planning problems is referred to as STRIPS. Following Geffner and Bonet [2013, p. 24], we have the following definition:

Definition 4 (STRIPS planning problem)**.** A *STRIPS planning problem* is a tuple $\langle F, I, A, G \rangle$ where F is a finite set of propositional symbols, $I \subseteq F$ represents the initial state, A is the finite set of actions, and $G \subseteq F$ represents the goal. Furthermore, an action $a \in A$ is represented by three sets of atoms: the "Add list" $\mathsf{Add}(a)$, the "Delete list" $\mathsf{Del}(a)$, and the "Precondition list" $\mathsf{Pre}(a)$.

A STRIPS problem $\langle F, I, A, G \rangle$ represents a planning problem $\langle \Sigma, s_0, S_G \rangle$ where $\Sigma = \langle S, A, \delta \rangle$ is such that $S = 2^F$, $s_0 = I$, $S_G = \{s \in S \mid G \subseteq s\}$, and δ is as follows:

$$\delta(s, a) = \begin{cases} (s \setminus \mathsf{Del}(a)) \cup \mathsf{Add}(a) & \text{if } \mathsf{Pre}(a) \subseteq s \\ \text{undefined} & \text{otherwise} \end{cases}$$

Given a set of fluents F, we will write $\mathcal{L}(F)$ to denote the set of Boolean formulas over those fluents, i.e., formulas constructed using negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), and so on as usual. A *literal* is either an atomic formula $f \in F$ or its negation $\neg f$. We define the set of literals $\mathsf{lits}(F) = F \cup \{\neg f \mid f \in F\}$. Given a literal ℓ, we may write $\overline{\ell}$ for the complementary literal, i.e. $\overline{f} = \neg f$ and $\overline{\neg f} = f$.

Given a STRIPS state $s \subseteq F$ and formula $\varphi \in \mathcal{L}(F)$, we will write $s \models \varphi$ if φ is true under the truth assignment which maps all the fluents in s to true and all the fluents in $F \setminus s$ to false. Sometimes we will treat a set $\varphi \subseteq F$ of atoms as their conjunction.

3 Minimizing (Negative) Side Effects

In this section we formalize several versions of side-effect-minimizing objectives for use in classical planning.

As suggested informally in Section 1, a side effect of a plan is any change to the state, resulting from the execution of the plan, that was not prescribed explicitly as part of the goal. We define a plan that minimizes change as follows.

Definition 5 (Side-effect minimizing plan)**.** Given a planning problem $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ where $\Sigma = \langle S, A, \delta \rangle$, and a *distance function* $d : S \times S \to [0, \infty)$, a plan π for $\mathcal{P}$ is side-effect minimizing iff there is no plan π' for $\mathcal{P}$ such that $d(\delta(s_0, \pi'), s_0) < d(\delta(s_0, \pi), s_0)$.

The distance function is a proxy for measuring the difference or change between two states – above, the goal state and the initial state – and our objective is to minimize such change. Depending on how the distance function is defined, it can be used to minimize *all* change or change with additional qualifications. (Despite it being called "distance", the function does not have to be symmetric.)

The side-effect minimizing objective in Definition 5 could be viewed as a classical planning version of the "very naive approach" to side effects discussed by Amodei *et al.* [2016]:

> A very naive approach would be to penalize state distance $d(s_i, s_0)$ between the present state s_i and some initial state s_0. Unfortunately, such an agent wouldn't just avoid changing the environment—it will resist any other source of change, including the natural evolution of the environment and the actions of any other agents!

This "naivety" does not stand out so much in our context, since a standard assumption of classical planning "excludes the possibility of actions by other actors, or exogenous events that are not due to any actor" [Ghallab *et al.*, 2016, p. 20].

Note that Definition 5 minimizes the distance (the change) between the initial state and the goal state – the state of the world after the plan has been executed. It is often important to consider to what extent intermediate states realized during the execution of a plan deviate from the initial state. Consider, for example, a plan to buy groceries. It may be more efficient to unlock and open the house door, exit the house, go to the store, buy groceries, return home, enter the house, and then close and lock the door, having left the door to the house open (and vulnerable to other events) for much of the duration of the plan, restoring it to its locked state at the end of the outing. In such cases it might make sense to favour plans that minimize the duration of side effects by aggregating the distances between the initial state and each successive state on the path of the plan, as shown below. Such a criterion could be used on its own or to further distinguish equally preferred side-effect minimizing plans following Definition 5.

Definition 6 (Within-plan side effect minimization)**.** Given a planning problem $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ where $\Sigma = \langle S, A, \delta \rangle$, and a *distance function* $d : S \times S \to [0, \infty)$, a plan $a_1, \ldots, a_k \in A^*$ for $\mathcal{P}$ is within-plan-side-effect minimizing iff there is no plan $b_1, \ldots, b_\ell \in A^*$ for $\mathcal{P}$ such that

$$\sum_{i=1}^{\ell} d(\delta(s_0, b_1, \ldots, b_i), s_0) < \sum_{i=1}^{k} d(\delta(s_0, a_1, \ldots, a_i), s_0).$$

For the purposes of this paper, we will focus on side effects that hold *after* the execution of a plan, per Definition 5.

3.1 Minimizing Side Effects in STRIPS

In the case where a planning problem is represented in STRIPS, having a vocabulary of fluents gives an obvious way to define side effects and measure distance between states.

Definition 7 (STRIPS side effects)**.** Let $\langle F, I, A, G \rangle$ be a STRIPS planning problem, and π a plan that solves it. Then $f \in F$ is a *side effect* of π if $f \in \delta(I, \pi)$ but $f \notin I$ and $f \notin G$. Furthermore, if $f \notin \delta(I, \pi)$ but $f \in I$, we will say the literal $\neg f$ is a side effect of π.

Intuitively, a literal is a side effect of π if π changes the literal's truth value even though the goal didn't specify that the truth value should change. Side effects are not necessarily spurious. They are often causally necessary to achievement of the goal via a particular plan. However they may not be necessary to the achievement of a goal in all cases. We could distinguish between *necessary* side effects in relation to a planning problem – those that occur in *every* plan to achieve the goal G, and contrast these with *discretionary* side effects – those that arise and that may (or may not) be causally necessary w.r.t. a particular subset of plans. Necessary side effects, and their relation to *landmarks* [Hoffmann *et al.*, 2004], are discussed further in Appendix A.1.

Definition 8 (The set $\mathsf{SideEffect}(\mathcal{P}, \pi)$)**.** Given a STRIPS planning problem $\mathcal{P} = \langle F, I, A, G \rangle$ and a plan $\pi \in A^*$, $\mathsf{SideEffect}(\mathcal{P}, \pi)$ denotes the set of literals that are side-effects of π w.r.t. $\mathcal{P}$ (as described in Definition 7).

One simple idea is to try to find a plan for a planning problem that minimizes the number of side effects.

Definition 9 (Side-effect minimizing plan (STRIPS version))**.** Given a STRIPS planning problem $\mathcal{P} = \langle F, I, A, G \rangle$ and associated plan π for $\mathcal{P}$, π is a *side-effect minimizing* plan iff there is no plan π' for $\mathcal{P}$ such that $|\mathsf{SideEffect}(\mathcal{P}, \pi')| < |\mathsf{SideEffect}(\mathcal{P}, \pi)|$.

This can be viewed as an instantiation of Definition 5 with a distance function defined so that $d(\delta(I, \pi), I) = |\mathsf{SideEffect}(\mathcal{P}, \pi)|$. It makes no attempt to distinguish side effects that are in any sense negative.

The determination of whether a side effect is negative or positive is often domain specific – perhaps characterized by the utility associated with the effect in question. For example, changing some fluents (e.g., one representing whether a vase is broken) could be given greater "weight" than changing others. Such domain-specific details could be built into a domain-tailored distance function. In the section that follows, we propose and characterize classes of negative side effects that relate, in general terms, to the impact an acting agent has on other agents, and are domain independent in the sense that they rely on general properties of the planning problem specification.

4 Effects of Plans on Other Agents

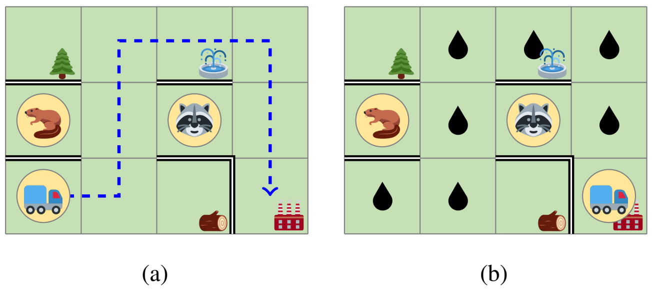
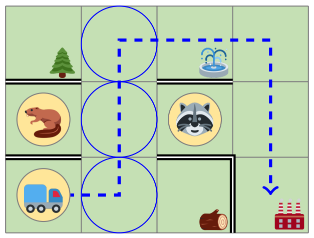
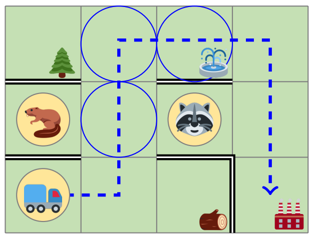


Figure 1: The robot truck (🚚), beaver (🦫), and raccoon (🦝) can move to adjacent cells, but cannot pass through walls (╗) or each other. The robot wants to get to the factory (🏭), but each cell it touches is contaminated with oil (💧), after which it cannot be visited by animals. If the robot just goes directly to the factory – following the blue path in (a) – then in the resulting state (b) the beaver is unable to reach the tree (🌲) or wood (🪵), and the raccoon is unable to go wash its hands in the fountain (⛲). If the robot can clean up contaminated cells, but has a limited budget for doing so, how should it modify its plan to minimize side effects on the animals?

To this point, our planning problems have implied the existence of a single agent – the *acting agent*, who is conceiving and executing plans. Here we introduce an environment in which there are multiple agents, co-existing in a shared environment – each with its own actions, goals and plans that it wishes to be able to execute at some point. We provide this environment to introduce classes of (negative) side effects.

Figure 1 introduces an example that we will use as an illustration throughout this section, in which a robot truck's actions may prevent other agents – here, animals – from achieving their goals or from following particular paths.


(a) By cleaning the circled cells, the robot will allow the beaver to reach either the tree or the wood.


(b) The robot will allow the beaver to reach the tree and the raccoon to reach the fountain.

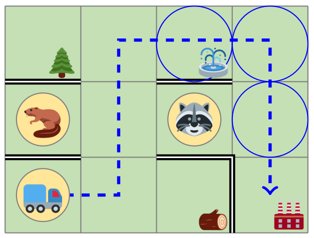

(c) The robot will allow the raccoon to reach the fountain by a path on the right edge.

Figure 2: In each image, the robot truck plans to follow the same blue path to the factory. The circled cells in each image are the ones that the robot will clean in that plan (after contaminating them).

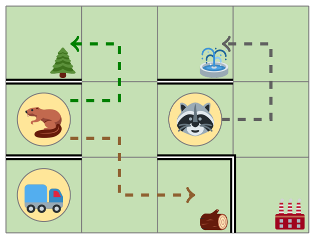


Figure 3: Possible animal plans: The beaver could go to the tree (green path) or the wood (brown path), and the raccoon could go to the fountain (gray path).

We claim that important classes of negative side effects are those that preclude agents operating in a shared environment from realizing possible goals or plans they might have been able to realize had the acting agent not executed its plan. We argue that the acting agent should therefore construct plans that minimize such negative side effects; i.e, that they should conceive and prefer plans that minimize the effect that the plans' execution will have on other agents' (and possibly their own) ability to achieve their goals or plans in the future.

In the rest of the section, we first describe the setup we will consider – a multiagent planning environment in which agents co-exist. This is followed by general definitions of the classes of negative side effects that correspond to our claim, along with a number of definitions of types of plans that in some sense minimize negative side effects.

Definition 10 (Multiagent planning environment)**.** A *multiagent planning environment* is a planning problem $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ where $\Sigma = \langle S, A, \delta \rangle$, as in Definition 2, but where the set of actions is partitioned as $A = \bigcup_{i=1}^{n} A_i$, where the subsets correspond to the actions available to each of the n distinct agents i. We designate agent 1 as the *acting agent*.

Despite the multiagent setting, our focus is solely on the plan of the acting agent, and how it can act to avoid causing negative side effects for agents who act subsequently. Functionally, this amounts to considering a simplified setting in which there are two phases: first, the acting agent executes a plan to accomplish its goal, and then afterwards some agent (possibly the same agent again) has the opportunity to execute an additional action sequence. By way of example, consider a set of roommates who share a kitchen. Each prepares their dinner alone, but with the understanding that any one of the roommates will subsequently use the kitchen to prepare a meal, and that the acting agent should minimize their negative impact on whoever uses the kitchen next.

Since the agents are operating in a shared state space, the actions of the first agent could affect others in a positive or negative way. A positive side effect might advance or simplify the future goals of other agents, while a negative side effect would impede other agents' goal or plan realization. The notion of constructing plans that have *positive* side effects could be an important aspiration, and is related to notions like *helpfulness* [Freedman *et al.*, 2020]. A treatment of positive side effects could be realized through a mirroring of many of the ideas put forward here; we will not elaborate further at this time.

The following definition will be useful.

Definition 11 (Achievable/Unachievable)**.** Given a multi-agent planning environment $\mathcal{P}$, a goal $S'_G \subseteq S$ is achievable by agent i in state $s \in S$, written $\mathsf{achievable}(S'_G, i, s)$, if there exists a plan $\pi' \in A_i^*$ (i.e., using agent i's actions) such that $\delta(s, \pi') \in S'_G$. If no such plan exists, then we say the goal is unachievable by agent i in s: $\mathsf{unachievable}(S'_G, i, s)$.

We can now define (a class of) negative side effects.

Definition 12 (Negative side effects (w.r.t. a goal))**.** Suppose $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ is a multi-agent planning environment. Suppose $S'_G \subseteq S$ is another goal and $\mathsf{achievable}(S'_G, i, s_0)$. Then a plan $\pi \in A_1^*$ for $\mathcal{P}$ has negative side effects on agent i w.r.t. goal S'_G if $\mathsf{unachievable}(S'_G, i, \delta(s_0, \pi))$.

Definition 12 captures the case where the acting agent's plan precludes the subsequent achievement of another agent's goal. There are many possible variants to this definition. If we were to add a quality measure (for example action costs or reward) then a negative side effect of a plan could be one that compromises the quality of another agent's subsequent achievement of their goal.

While it may be reasonable for the acting agent to be aware of the possible goals that other agents may wish to pursue in the future, it it is less reasonable to know which agent will act next or which of its goals it will pursue. As such, in the following, we provide the acting agent with goals corresponding to different agents, and task the acting agent with constructing a *goal-set preserving plan* – one that ensures that, following execution, as many of those goals as possible will be achievable by the associated agent.

Definition 13 (Goal-set preserving plan)**.** Given a multiagent planning environment $\mathcal{P}$ (with n agents) together with a set H of goal-agent pairs, $\langle S'_G, i \rangle$, where $S'_G \subseteq S$ and $i \in \{1, \ldots, n\}$ and $\mathsf{achievable}(S'_G, i, s_0)$, a *goal-set-preserving plan* for $\mathcal{P}$ with respect to H is a plan $\pi \in A_1^*$ which, among all such plans in A_1^*, maximizes the cardinality of the set

$$\{\langle S'_G, i \rangle \mid \langle S'_G, i \rangle \in H \text{ and } \mathsf{achievable}(S'_G, i, \delta(s_0, \pi))\}.$$

A goal-set preserving plan minimizes the number of goals (from the given set H) with respect to which it has negative side effects. To illustrate, consider the multiagent planning environment shown in Figure 1, and suppose the set H contains three possibilities: the beaver wants to go the tree, the beaver wants to go to the wood, and the raccoon wants to go to the fountain. Various possible plans for the robot truck are shown in Figure 2. The plans in Figures 2a and 2b allow for two of the possible goals in H to be achieved after the robot acts, and so are goal-set preserving if the robot cannot clean more than three cells (to allow all three goals to be achieved after it acts, the robot would have to clean four cells).

Now, suppose that the acting agent alternatively has information about what *plans* other agents might follow in pursuit of goals. We can define a variant of Definition 12 where a plan has negative side effects on agent i if i is not able to achieve its goal by the same plan that would have worked in s_0 (prior to the execution of the acting agent's plan):

Definition 14 (Negative side effects (w.r.t. a plan))**.** Suppose $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ is a multi-agent planning environment. Further suppose $S'_G \subseteq S$ is another goal and $\pi' \in A_i^*$ is a plan s.t. $\delta(s_0, \pi') \in S'_G$. Then a plan $\pi \in A_1^*$ for $\mathcal{P}$ has negative side effects on agent i w.r.t. goal S'_G and plan π' if $\delta(\delta(s_0, \pi), \pi') \notin S'_G$.

This commitment to a particular plan could be important to save agent i the effort of replanning, or if agent i is unaware of the acting agent's execution of their plan and that it has rendered i's original plan unachievable. A more sophisticated treatment of the latter point might involve some form of *epistemic planning* [Baral *et al.*, 2017].

We can now define a *plan-set preserving plan* in the spirit of Definition 13, given a set H of goal-plan pairs $\langle S'_G, \pi' \rangle$ where S'_G is a possible future goal and π' is a plan by which some agent might achieve that goal.

Definition 15 (Plan-set preserving plan)**.** Given a multiagent planning environment $\mathcal{P}$ together with a finite set H of goal-plan pairs $\langle S'_G, \pi' \rangle$, where $S'_G \subseteq S$, $\pi' \in A_i^*$ for some i, and $\mathsf{achievable}(S'_G, i, s_0)$, a *plan-set-preserving plan* for $\mathcal{P}$ with respect to H is a plan $\pi \in A_1^*$ which, among all such plans in A_1^*, maximizes the cardinality of the set

$$\{\langle S'_G, \pi' \rangle \mid \langle S'_G, \pi' \rangle \in H \text{ and } \delta(\delta(s_0, \pi), \pi') \in S'_G\}.$$

Note that in this definition, unlike Definition 13, it matters whether agents can still use the same plans to achieve goals that would have worked in s_0. A plan-set preserving plan minimizes the number of plans (from the given set H) with respect to which it has negative side effects (Definition 14).

To illustrate, consider again the example from Figure 1. Suppose that we have a set H of goal-plan pairs with the three possibilities illustrated by Figure 3. Then the robot truck's plan in Figure 2a is plan-set preserving if it cannot clean more than three cells. However, neither of the other plans shown in Figure 2 is plan-set preserving. While in Figure 2b the raccoon has a path to the fountain, that is not the path that would be traversed by the raccoon's plan in H.

Instead of just having a set of possible future goal-agent or goal-plan pairs, as in Definitions 13 and 15, we could make use of a probability distribution over what agent is next going to try to achieve what goal, or what agent is next going to try to follow what plan. This could be reasonably acquired from experience in many cases, since many goals commonly recur (e.g., library visitors are likely to want to check out a book).

Definition 16 (Goal-preserving plan)**.** Suppose $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ is a multiagent planning environment with n agents and $\mathsf{P_G}$ is a probability function assigning probabilities to subsets of the sample space of goal-agent pairs,

$$\mathcal{H} = \{\langle S'_G, i \rangle \mid S'_G \subseteq S \text{ and } i \in \{i, \ldots, n\} \text{ and } \\ \mathsf{achievable}(S'_G, i, s_0)\}.$$

Then a *goal-preserving plan* for $\mathcal{P}$ with respect to $\mathsf{P_G}$ is a plan $\pi \in A_1^*$ which, among all such plans in A_1^*, maximizes the probability

$$\mathsf{P_G}(\{\langle S'_G, i \rangle \in \mathcal{H} \mid \mathsf{achievable}(S'_G, i, \delta(s_0, \pi))\}),$$

i.e., the sum

$$\sum_{\langle S'_G, i \rangle \in \mathcal{H} : \mathsf{achievable}(S'_G, i, \delta(s_0, \pi))} \mathsf{P_G}(\langle S'_G, i \rangle).$$

That is, a goal-preserving plan maximizes the probability that whichever agent acts after the first agent can still achieve its goal (in other words, it minimizes the probability of having negative side effects on the agent that acts next w.r.t. its goal). Recall that environments are deterministic, so the only uncertainty is about who will act and what their goal is.

Observe that if $\mathsf{P_G}$ assigns all the probability mass uniformly over some subset H of the sample space $\mathcal{H}$, then a goal-preserving plan is equivalent to a goal-set preserving plan. So goal-preserving plans can be seen as a generalization of goal-set preserving plans.

We can also have a probabilistic version of Definition 15:

Definition 17 (Plan-preserving plan)**.** Suppose $\mathcal{P} = \langle \Sigma, s_0, S_G \rangle$ is a multiagent planning environment and $\mathsf{P_P}$ is a probability function assigning probabilities to subsets of the sample space of goal-plan pairs,

$$\mathcal{H} = \{\langle S'_G, \pi' \rangle \mid S'_g \subseteq S \text{ and } \pi' \in A_i^* \text{ for some } i \text{ and } \\ \delta(s_0, \pi') \in S'_G\}.$$

Then, given $\mathsf{P_P}$, a *plan-preserving plan* for $\mathcal{P}$ is a plan $\pi \in A_1^*$ for $\mathcal{P}$ which, among all such plans in A_1^*, maximizes the probability

$$\mathsf{P_P}(\{\langle S'_G, \pi' \rangle \in \mathcal{H} \mid \delta(\delta(s_0, \pi), \pi') \in S'_G)\}).$$

Plan-preserving plans can also be seen as a generalization of plan-set preserving plans.

Summary. Table 1 summarizes the different objectives relating to avoiding side effects that we have considered. Note that all the later definitions – other than Definition 6 – could be thought of as instantiations of Definition 5 with appropriate distance functions. For example, for Definition 13, the distance function would just have to be such that $d(s, s_0) = |\{\langle S'_G, i \rangle \in H \mid \mathsf{unachievable}(S'_G, i, s)\}|$.

5 Computing Side-Effect Minimizing Plans

In this section we address the problem of how to compute the types of side-effect minimizing plans previously characterized. For the purposes of this paper, we assume the planning

Def.	Additional information used	Objective
5	State distance function	Minimize distance of final state from initial state
6	State distance function	Minimize sum of distances of traversed states from initial state
9	None (other than a STRIPS representation)	Minimize number of fluents changed
13	Set of possible future goal-agent pairs	Maximize number of future goals achievable by corresponding agent
15	Set of possible future goal-plan pairs	Maximize number of future goals achievable by corresponding plan
16	Distribution over possible future goal-agent pairs	Maximize probability future goal can be achieved by corresponding agent
17	Distribution over possible future goal-plan pairs	Maximize probability future goal can be achieved with corresponding plan

Table 1: Summary of different side-effect-minimizing objectives

problems are represented as STRIPS problems, so we will be using G instead of S_G to refer to the goal, and G' instead of S'_G to refer to a possible future goal. So for example, if we're trying to find a plan-preserving plan for a STRIPS problem $\langle F, I, A, G \rangle$, we will have a distribution P_P over pairs $\langle G', \pi' \rangle$ where $G' \subseteq F$.

We show how to find plans that are side-effect minimizing (Definition 9), plan-preserving (Definition 17), or goal-preserving (Definition 16) by compiling the STRIPS problem into a STRIPS problem with action costs and soft goals. As previously noted, plan-set-preserving plans (Definition 15) can be viewed as a special case of plan-preserving plans, and goal-set-preserving plans (Definition 13) as a special case of goal-preserving plans. Therefore, our approaches will also handle those plan concepts. We leave minimizing within-plan side effects (Definition 6) to future work.

5.1 Background

In this subsection we provide technical definitions and notation necessary to the approach that follows.

Regression. Regression can be thought of as a form of pre-image computation. It is a rewriting operation, first introduced by Waldinger [1975], that given an action a and a state, s', resulting from performing a, returns a formula that characterizes the conditions that must have been true in the previous state to result in s'. (In this regard, the formula parsimoniously characterizes a family of states.) Following Muise [2014, Definition 2], we have the following definition:

Definition 18 (Regression in STRIPS)**.** Given a STRIPS problem $\langle F, I, A, G \rangle$, if $\varphi \subseteq F$ and $a \in A$, then the regression of φ through a, written $\mathcal{R}(\varphi, a)$, is defined as follows:

$$\mathcal{R}(\varphi, a) \overset{\text{def}}{=} \begin{cases} (\varphi \setminus \mathsf{Add}(a)) \cup \mathsf{Pre}(a) & \text{if } \mathsf{Del}(a) \cap \varphi = \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$

Note that the result of regression is not interpreted as a state but as representing a set of states (the set of states that make it true). We use $\mathcal{R}^*(\varphi, \pi)$ to denote the iterated regression through all the actions in π in order.

The significance of regression for us comes from the following result, which is a specialization of Reiter's regression theorem [Reiter, 2001].

Theorem 1 (Regression Theorem for STRIPS [Muise, 2014, Theorem 2])**.** *An action sequence $\vec{a}$ is a plan for a STRIPS planning problem $\langle F, I, A, G \rangle$ if and only if $I \models \mathcal{R}^*(G, \vec{a})$.*

We find it useful to rephrase that in a more general way:

Corollary 1. *Given a STRIPS planning problem $\langle F, I, A, G \rangle$, a state $s \in 2^F$, an action sequence $\vec{a}$, and a set of atoms $\varphi \subseteq F$, $\delta(s, \vec{a}) \models \varphi$ if and only if $s \models \mathcal{R}^*(\varphi, \vec{a})$.*

Proof. Apply Theorem 1 to $\langle F, s, A, \varphi \rangle$. □

Planning with soft goals and action costs. As noted above our strategy to compute different classes of side-effect-minimizing plans is to compile them into a planning problem with soft goals and action costs. The following definition is based on Keyder and Geffner's [2009, Definition 2], but incorporates a more general utility function, assigning utilities to formulas instead of fluents, as Keyder and Geffner [2009, section 4] later suggest.

Definition 19 (STRIPS problem with action costs and soft goals)**.** A STRIPS problem with action costs and soft goals is a tuple $\langle F, I, A, G, c, u \rangle$ where $\langle F, I, A, G \rangle$ is a STRIPS problem, $c : A \to [0, \infty)$ is the action cost function, and $u : \mathcal{L}(F) \to [0, \infty)$ is the utility function.

Given a STRIPS problem with action costs and soft goals $\mathcal{P} = \langle F, I, A, G, c, u \rangle$, we call G the *hard goal* of $\mathcal{P}$, and any formula φ for which $u(\varphi) > 0$ a *soft goal*. For such a problem $\mathcal{P}$, the *utility* of a plan $\pi = a_1, \ldots, a_k$ is given by

$$u(\pi) = \left(\sum_{\varphi \in \mathcal{L}(F) : \delta(I, \pi) \models \varphi} u(\varphi) \right) - \sum_{i=1}^{k} c(a_i)$$

That is, the utility is the sum of the utilities of the soft goals satisfied in the state resulting from executing the plan, minus the sum of the action costs. An optimal plan will maximize utility (while achieving the hard goal). One way to solve STRIPS problems with action costs and soft goals is to compile the soft goals away, as shown by Keyder and Geffner [2009], and to use an off-the-shelf cost-optimal planner.

Definition 20 ($\mathsf{Con}(\cdot)$)**.** Given a set of fluents $F' \subseteq F$, we will use $\mathsf{Con}(F')$ to denote the conjunction of the fluents in F', in some canonical order.

Later, it will be important that $\mathsf{Con}(F')$ is equal to a unique formula, which is why we're not just using the notation $\bigwedge F'$.

5.2 Computing Side-Effect Minimizing Plans

Suppose we want to compute a side-effect minimizing plan (Definition 9) for a STRIPS problem $\mathcal{P} = \langle F, I, A, G \rangle$. We can do so by compiling $\mathcal{P}$ into a STRIPS problem with action costs and soft goals $\tilde{\mathcal{P}}$ as described below.

Definition 21 (Side-effect-minimizing compilation)**.** Given a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$, the side-effect-minimizing compilation of $\mathcal{P}$ is a STRIPS problem with action costs and soft goals $\tilde{\mathcal{P}} = \langle F, I, A, G, c, u\rangle$ where

- $c(a) = 0$ for all $a \in A$
- $u(\varphi) = \begin{cases} 1 & \text{if } \varphi \in I \cup \{\neg f \mid f \notin (I \cup G)\} \\ 0 & \text{otherwise} \end{cases}$

Proposition 1 (Correctness of side-effect-minimizing compilation)**.** *Given a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$ and its side-effect-minimizing compilation $\tilde{\mathcal{P}} = \langle F, I, A, G, c, u\rangle$, an action sequence $\vec{a} \in A^*$ is a side-effects minimizing plan for $\mathcal{P}$ if and only if $\vec{a}$ is an optimal plan for $\tilde{\mathcal{P}}$.*

Proof. See Appendix A.2 in the supplementary material. □

5.3 Computing Plan-Preserving Plans

How to compute plan-preserving plans? We consider the case where the given probability function over future goal-plan pairs P_P has finite support, i.e., it only assigns non-zero probabilities to finitely many pairs. Our approach is to compile the problem into a STRIPS problem with action costs and soft goals. To construct this, we take every goal-plan pair $\langle G', \pi'\rangle$ that has non-zero probability, and make the regression of G' through π' a soft goal. This is described in full detail below.

Definition 22 (Plan-preserving-plan compilation)**.** Suppose that we have a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$, where $A = A_1 \cup \cdots \cup A_n$, and a probability function P_P over pairs $\langle G', \pi'\rangle$ with finite support. We write the pairs that have non-zero probability under P_P as $\langle G_1, \pi_1\rangle, \ldots, \langle G_m, \pi_m\rangle$. We say that the *plan-preserving-plan compilation* of $\mathcal{P}$ is the STRIPS problem with action costs and soft goals $\tilde{\mathcal{P}} = \langle F, I, \tilde{A}, G, c, u\rangle$ where

- $\tilde{A} = A_1$
- $c(a) = 0$ for all $a \in \tilde{A}$
- $u(\varphi) = \sum_{j \in \{1,\ldots,m\} : \varphi = \mathsf{Con}(\mathcal{R}^*(G_j, \pi_j))} \mathsf{P}_\mathsf{P}(\langle G_j, \pi_j\rangle)$, where the regression operator is defined with respect to the original problem $\mathcal{P}$

Note that the definition of the utility function u in Definition 22 sums over goal-plan pairs, since there may be distinct pairs $\langle G_k, \pi_k\rangle$ and $\langle G_\ell, \pi_\ell\rangle$ for which $\mathsf{Con}(\mathcal{R}^*(G_k, \pi_k)) = \mathsf{Con}(\mathcal{R}^*(G_\ell, \pi_\ell))$, which increases the value of making that formula true. Note also that it's important that $\mathsf{Con}(\cdot)$ writes the conjuncts in a specific order (so that, e.g., we don't count the formulas $(p \wedge q)$ and $(q \wedge p)$ as different soft goals).

Proposition 2 (Correctness of plan-preserving-plan compilation)**.** *Given a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$ where $A = A_1 \cup \cdots \cup A_n$, and a probability function P_P over pairs $\langle G', \pi'\rangle$ (such that π' can achieve G' from I) with finite support, a sequence of actions $\vec{a} \in A_1^*$ is a plan-preserving plan for $\mathcal{P}$ just in case it is an optimal plan for the plan-preserving-plan compilation $\tilde{\mathcal{P}}$.*

Proof. See Appendix A.3 in the supplementary material. □

The assumption that P_P has finite support is not very constraining, since for it to *not* hold would require that non-zero probability be assigned to plans that were arbitrarily longer than they needed to be (P_P could have infinite support only by assigning non-zero probability to infinitely many plans for some particular goal). Furthermore, approximations of plan-preserving plans could be found by considering in the compilation only some number of the most probable pairs $\langle G', \pi'\rangle$.

5.4 Computing Goal-Preserving Plans

Suppose that you have a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$ (where $A = A_1 \cup \cdots \cup A_n$) and a probability function P_G over pairs $\langle G', i\rangle$ with achievable goals, and you want to find a goal-preserving plan. Since the sets of possible goals and agents are finite, there are only finitely many pairs that have non-zero probability under P_G, and we can write them as $\langle G_1, i_1\rangle, \ldots, \langle G_m, i_m\rangle$.

The intuition is that we want to convert $\mathcal{P}$ and P_G into a STRIPS problem with action costs and soft goals $\tilde{\mathcal{P}}$ in which every one of the G_j's is a soft goal. $\tilde{\mathcal{P}}$ will be set up so that an optimal plan will have a prefix of actions by the first agent, that will correspond to a goal-preserving plan for $\mathcal{P}$. Since for a goal-preserving plan for $\mathcal{P}$ we want G_j's to be possible to complete from the state resulting from the first agent's actions, in $\tilde{\mathcal{P}}$ we will have m additional copies of the state space. The first agent's actions to achieve G will modify every copy of the state in parallel. Then the soft goals G_j can (potentially) be achieved in sequence, each in a different copy of the state so that they don't interfere with each other. We'll introduce extra bookkeeping fluents $q_0, q_1, \ldots, q_m$ to keep track of which hard or soft goal is currently being worked on, and extra actions $switch_1, \ldots, switch_m$ that advance to the next goal. The full details are in the definition below.

Definition 23 (Goal-preserving-plan compilation)**.** Suppose we have a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$, where $A = A_1 \cup \cdots \cup A_n$, and a probability function P_G over pairs $\langle G', i\rangle$. We write the pairs that have non-zero probability under P_G as $\langle G_1, i_1\rangle, \ldots, \langle G_m, i_m\rangle$. We say that the *goal-preserving-plan compilation* of $\mathcal{P}$ is the STRIPS problem with action costs and soft goals $\tilde{\mathcal{P}} = \langle \tilde{F}, \tilde{I}, \tilde{A}, \tilde{G}, c, u\rangle$ where

- $\tilde{F} = \{f_j \mid f \in F \text{ and } 0 \leq j \leq m\} \cup \{q_0, \ldots, q_m\}$, where the q_j are new symbols not appearing in F
- $\tilde{I} = \{f_j \mid f \in I \text{ and } 0 \leq j \leq m\} \cup \{q_0\}$
- $\tilde{A} = \tilde{A}_\mathsf{plan} \cup \tilde{A}_\mathsf{future} \cup \tilde{A}_\mathsf{advance}$, where the components are as described below. For each action $a \in A_1$, $\tilde{A}_\mathsf{plan}$ includes a corresponding action $\tilde{a}$ where

$$\mathsf{Pre}(\tilde{a}) = \{f_0 \mid f \in \mathsf{Pre}(a)\} \cup \{q_0\}$$
$$\mathsf{Add}(\tilde{a}) = \{f_j \mid f \in \mathsf{Add}(a) \text{ and } 0 \leq j \leq m\}$$
$$\mathsf{Del}(\tilde{a}) = \{f_j \mid f \in \mathsf{Del}(a) \text{ and } 0 \leq j \leq m\}$$

For $j \in \{1, \ldots, m\}$, for each action $a \in A_{i_j}$, $\tilde{A}_\mathsf{future}$ includes an action $\tilde{a}^j$ where

$$\mathsf{Pre}(\tilde{a}^j) = \{f_j \mid f \in \mathsf{Pre}(a)\} \cup \{q_j\}$$
$$\mathsf{Add}(\tilde{a}^j) = \{f_j \mid f \in \mathsf{Add}(a)\}$$
$$\mathsf{Del}(\tilde{a}^j) = \{f_j \mid f \in \mathsf{Del}(a)\}$$

For $j \in \{1, \ldots, m\}$, $\tilde{A}_{\mathsf{advance}}$ includes an action $switch_j$ where $\mathsf{Pre}(switch_j) = \{q_{j-1}\}$, $\mathsf{Add}(switch_j) = \{q_j\}$, and $\mathsf{Del}(switch_j) = \{q_{j-1}\}$.

- $\tilde{G} = \{f_0 \mid f \in G\}$
- $c(a) = 0$ for all $a \in \tilde{A}$
- $u(\varphi) = \sum_{\substack{j \in \{1,\ldots,m\}: \\ \varphi = \mathsf{Con}(\{f_j \mid f \in G_j\})}} \mathsf{P_G}(\langle G_j, i_j \rangle)$

So the soft goals of the goal-preserving-plan compilation are of the form $\mathsf{Con}(\{f_j \mid f \in G_j\})$ – which we can read informally as G_j within the jth copy of the state space – for each j. (Note that except possibly if φ is True, the sum in the definition of $u(\varphi)$ will only have at most one summand.)

Observe how the use of the q_j fluents ensures that any plan for $\tilde{\mathcal{P}}$ will start with (0 or more) actions from $\tilde{A}_{\mathsf{plan}}$, possibly followed by the action $switch_1$ and then (0 or more) actions of the form $\tilde{a}^1 \in \tilde{A}_{\mathsf{future}}$, possibly followed by $switch_2$ and then (0 or more) actions of the form $\tilde{a}^2 \in \tilde{A}_{\mathsf{future}}$, and so on.

Proposition 3 (Correctness of goal-preserving-plan compilation)**.** *Given a STRIPS problem $\mathcal{P} = \langle F, I, A, G \rangle$ where $A = A_1 \cup \cdots \cup A_n$, and a probability function $\mathsf{P_G}$ over pairs $\langle G', i \rangle$ (such that agent i can achieve G' from I), a sequence of actions $a_1, \ldots, a_k \in A_1^*$ is a goal-preserving plan for $\mathcal{P}$ just in case there is an optimal plan for the goal-preserving-plan compilation $\tilde{\mathcal{P}}$ that has as a prefix the corresponding sequence of actions $\tilde{a}_1, \ldots, \tilde{a}_k \in \tilde{A}^*_{\mathsf{plan}}$, and any actions following those are not in $\tilde{A}^*_{\mathsf{plan}}$.*

Proof. See Appendix A.4 in the supplementary material. □

6 Discussion and Related Work

There are a number of approaches to handling side effects in MDPs. Some involve interacting with a human to get further information about what side effects are negative [Zhang *et al.*, 2018; Saisubramanian *et al.*, 2020a]. More related to the present work are those not making use of human feedback.

In particular, our work was inspired by approaches in which an auxiliary reward is introduced into an MDP to encourage the agent to preserve its own ability to reach states [Krakovna *et al.*, 2019], gain reward from other reward functions [Turner *et al.*, 2020], or complete tasks from a given distribution [Krakovna *et al.*, 2020]. A key difference of our approaches is that we allow for considering more than one agent's abilities. Krakovna *et al.* [2020] wrote that "Our main insight is that side effects matter because we may want the agent to perform other tasks after the current task in the same environment." Our work can be viewed as generalizing that insight to also consider that *other* agents (who may have different actions available to them) may want to perform other tasks after the first agent in the same environment. Note that Turner [2019] did informally discuss considering other agents' attainable utilities.

Within the field of planning, one of the more related papers is by Freedman and Zilberstein [2017]. Their definition of *independent interaction* (their Definition 3) includes that agent "R_{ing} has a personal goal G' to accomplish, but should avoid preventing [agent] R_{ed} from accomplishing her own task at the same time." That sounds much like some of the things we've been discussing in this paper; however, their formalization is rather different. They formalize independent interaction as a planning problem in which the agents act in parallel to collectively achieve both R_{ing}'s goal G' and also all properties that might be desired by R_{ed} (with probability higher than a threshold).

In the context of *active goal recognition*, in which an observing agent performs actions to try to determine the goal G of another agent from a set $\mathcal{G}$ of hypothesized possible goals, Shvo and McIlraith [2020] deem an observer's plan τ to be *non-intervening* "if for every hypothesis $G \in \mathcal{G}$ the set of plans π, whose execution achieves G are preserved under the execution of τ." This could be thought of as the observer not having side effects on the observed agent.

Limitations. For the most part, in this paper, we considered side effects to be properties of the terminating state of the acting agent's plan (the exceptions are Definition 6, and Definition 25 in Appendix A.1). As a consequence, we favoured plans whose side effects minimized the impact on whatever goal (or plan) was going to be pursued immediately following the execution of the acting agent's plan. This presents two issues: first, this treatment of side effects is somewhat myopic in that it doesn't look further into the future; second, it does not allow for considering side effects in a true multiagent environment where other agents are operating simultaneously alongside the acting agent [Alizadeh Alamdari *et al.*, 2021].

Another issue in dealing with side effects that the methods proposed in this paper are not designed to address is having a limited vocabulary. Saisubramanian *et al.* [2020b] noted that a limited state representation "potentially affects the process of learning to avoid negative side effects." We're not doing learning, but our side-effect minimizing definitions rely on being able to make relevant state distinctions. Unfortunately, if (for example) the goal doesn't mention that the vase shouldn't be broken, then it may be that the language was designed so it's not possible to represent whether it's broken.

7 Conclusion

In this paper, we have considered the problem of avoiding negative side effects in classical planning. We have presented several versions of side-effect-minimizing objectives, and showed how, in the case of STRIPS planning problems, to compute them through compilations into STRIPS problems with soft goals.

There are many avenues for future work. Different side-effect-minimizing objectives could be defined, taking into account other aspects like action costs (e.g., how does the acting agent's plan change the cost of other agents achieving their goals?). Algorithmically, the compilations we have presented are quite straight-forward and alternative techniques for computing side-effect-minimizing plans could be investigated. Finally, the idea in this paper of considering the effect of actions on other agents' abilities is one that might be useful to bring from planning to reinforcement learning (we have made an attempt to do so [Alizadeh Alamdari *et al.*, 2021]).

Acknowledgements

We gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada CIFAR AI Chairs Program, and Microsoft Research. Finally, we thank the Schwartz Reisman Institute for Technology and Society for providing a rich multidisciplinary research environment.

The emojis in this paper are from the Twitter Emoji library (https://github.com/twitter/twemoji) and are copyright Twitter, Inc and other contributors, licensed under CC-BY 4.0 (https://creativecommons.org/licenses/by/4.0/). The truck and droplet emojis were modified.

References

[Alizadeh Alamdari *et al.*, 2021] Parand Alizadeh Alamdari, Toryn Q. Klassen, Rodrigo Toro Icarte, and Sheila A. McIlraith. Be considerate: Objectives, side effects, and deciding how to act. *arXiv preprint arXiv:2106.02617*, 2021.

[Amodei *et al.*, 2016] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[Baral *et al.*, 2017] Chitta Baral, Thomas Bolander, Hans van Ditmarsch, and Sheila McIlraith. Epistemic Planning (Dagstuhl Seminar 17231). *Dagstuhl Reports*, 7(6):1–47, 2017.

[Freedman and Zilberstein, 2017] Richard G. Freedman and Shlomo Zilberstein. Integration of planning with recognition for responsive interaction using classical planners. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4581–4588, 2017.

[Freedman *et al.*, 2020] Richard G. Freedman, Steven J. Levine, Brian C. Williams, and Shlomo Zilberstein. Helpfulness as a key metric of human-robot collaboration. *arXiv preprint arXiv:2010.04914*, 2020.

[Geffner and Bonet, 2013] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013.

[Ghallab *et al.*, 2004] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated Planning Theory and Practice*. Elsevier, 2004.

[Ghallab *et al.*, 2016] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016.

[Hoffmann *et al.*, 2004] Jörg Hoffmann, Julie Porteous, and Laura Sebastia. Ordered landmarks in planning. *Journal of Artificial Intelligence Research*, 22:215–278, 2004.

[Keyder and Geffner, 2009] Emil Keyder and Hector Geffner. Soft goals can be compiled away. *Journal of Artificial Intelligence Research*, 36:547–556, 2009.

[Krakovna *et al.*, 2019] Victoria Krakovna, Laurent Orseau, Miljan Martic, and Shane Legg. Penalizing side effects using stepwise relative reachability. In *Proceedings of the Workshop on Artificial Intelligence Safety 2019 co-located with the 28th International Joint Conference on Artificial Intelligence, AISafety@IJCAI 2019*, volume 2419 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

[Krakovna *et al.*, 2020] Victoria Krakovna, Laurent Orseau, Richard Ngo, Miljan Martic, and Shane Legg. Avoiding side effects by considering future tasks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[McCarthy and Hayes, 1969] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[McCarthy, 1977] John McCarthy. Epistemological problems of artificial intelligence. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 1038–1044, 1977.

[Muise, 2014] Christian Muise. *Exploiting Relevance to Improve Robustness and Flexibility in Plan Generation and Execution*. PhD thesis, University of Toronto, 2014.

[Reiter, 2001] Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[Saisubramanian *et al.*, 2020a] Sandhya Saisubramanian, Ece Kamar, and Shlomo Zilberstein. A multi-objective approach to mitigate negative side effects. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 354–361, 2020.

[Saisubramanian *et al.*, 2020b] Sandhya Saisubramanian, Shlomo Zilberstein, and Ece Kamar. Avoiding negative side effects due to incomplete knowledge of AI systems. *arXiv preprint arXiv:2008.12146*, 2020.

[Shvo and McIlraith, 2020] Maayan Shvo and Sheila A. McIlraith. Active goal recognition. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 9957–9966, 2020.

[Turner *et al.*, 2020] Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. Conservative agency via attainable utility preservation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pages 385–391, 2020.

[Turner, 2019] Alex Turner. Reframing impact. Blog post, https://www.lesswrong.com/s/7CdoznhJaLEKHwvJW, 2019.

[Waldinger, 1975] Richard Waldinger. Achieving several goals simultaneously. Technical Note 107, SRI Project 2245, 1975.

[Zhang *et al.*, 2018] Shun Zhang, Edmund H. Durfee, and Satinder P. Singh. Minimax-regret querying on side effects for safe optimality in factored Markov decision processes. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4867–4873, 2018.

A Supplementary Material

This supplementary material is structured as follows. First, Appendix A.1 elaborates on the discussion of *necessary* side effects from subsection 3.1. Then, Appendices A.2, A.3, and A.4 provide proofs of the propositions from section 5 about the correctness of the various plan compilations. Specifically, Appendix A.2 is about side-effect minimizing plans (proving Proposition 1), Appendix A.3 is about plan-preserving plans (proving Proposition 2), and Appendix A.4 is about goal-preserving plans (proving Proposition 3).

A.1 Necessary Side Effects and Landmarks

In subsection 3.1 we distinguished between necessary and discretionary side effects, noting that there was a relationship to landmarks, a property of planning problems that has proven useful in the construction of heuristics for automated plan generation. We begin by reviewing the definition of a landmark (following, e.g., Hoffmann *et al.* [2004]).

Definition 24 (Landmark)**.** Given a STRIPS planning problem $\langle F, I, A, G\rangle$, a formula $\varphi \in \mathcal{L}(F)$ is a landmark if φ holds in some state s on the solution path of every plan for $\langle F, I, A, G\rangle$.

According to this definition, the goal formula and more specifically the individual fluents that comprise the goal are all landmarks, as are intermediate subgoals that exist in every plan to achieve the goal.

In this paper, in the context of STRIPS, we generally deemed side effects to be those properties that hold in the *terminating state* of the acting agent's plan that are not part of the goal and that did not hold in the initial state (see Definition 7). The exception to this was our brief discussion of within-plan side effects associated with Definition 6 (within-plan side effect minimization). In particular, we discussed the importance of considering side effects that are realized during the execution of the plan and that may or may not persist until the end of the plan. We argued that minimizing the duration of time a side effect holds can be important when considering such within-plan side effects.

In the context of our discussion of necessary side effects and landmarks, it's interesting to further distinguish the notion of a STRIPS within-plan side effect, as follows:

Definition 25 (STRIPS within-plan side effect)**.** Let $\langle F, I, A, G\rangle$ be a STRIPS planning problem, and π a plan that solves it. Then $\ell \in \mathsf{lits}(F)$ is a *within-plan side effect* of π if $I \not\models \ell$ and $\ell \notin G$ but there exists a state s on the plan path from I to $\delta(I, \pi)$ such that $s \models \ell$.

With this additional notion of side effect defined, we define the notion of a necessary within-plan side effect for a STRIPS planning problem.

Definition 26 (Necessary within-plan side effect (STRIPS))**.** A necessary within-plan side effect for a STRIPS planning problem $\mathcal{P} = \langle F, I, A, G\rangle$ is a literal $\ell \in \mathsf{lits}(F)$ that is a within-plan side effect of π for every plan π that solves $\mathcal{P}$.

The following propositions follow straightforwardly from the definitions above.

Proposition 4. *If a literal $\ell \in \mathsf{lits}(F)$ is a necessary within-plan side effect of a STRIPS planning problem $\mathcal{P} = \langle F, I, A, G\rangle$, then ℓ is a landmark of $\mathcal{P}$.*

Proposition 5. *Given a STRIPS planning problem $\mathcal{P} = \langle F, I, A, G\rangle$, if a literal ℓ is a landmark of $\mathcal{P}$ such that $I \not\models \ell$ and $\ell \notin G$ then ℓ is a necessary within-plan side effect of $\mathcal{P}$.*

The relationship between landmarks and necessary within-plan side effects is interesting because the computation of landmarks is well understood and routinely used for heuristics in planning. As such it could be used as a means of generating necessary within-plan side effects. Importantly, in the context of the work presented in this paper, recognition of an undesirable or unsafe necessary within-plan side effect – an undesirable landmark – might suggest that the plan not be pursued at all. This presents a way to potentially verify violation of a safety constraint.

A.2 Proof of Proposition 1

First, we will find it useful to define the set of "potential side effects" of a problem,

$$\mathsf{PSE}(\mathcal{P}) = \{f \in F \mid f \notin I \cup G\} \cup \{\neg f \mid f \in I\}.$$

Note that $\mathsf{PSE}(\mathcal{P})$ includes every literal that could be in $\mathsf{SideEffect}(\mathcal{P}, \pi)$ for any π, and that the set $\mathsf{SideEffect}(\mathcal{P}, \pi)$ could be written as

$$\{\varphi \in \mathsf{PSE}(\mathcal{P}) \mid \delta(I, \pi) \models \varphi\}.$$

Now, the utility of a plan π for $\tilde{\mathcal{P}}$ is

$$\begin{aligned} u(\pi) &= \left(\sum_{\varphi \in \mathcal{L}(F) : \delta(I,\pi) \models \varphi} u(\varphi)\right) - \sum_{a \in \pi} c(a) \\ &= \left(\sum_{\varphi \in I \cup \{\neg f \mid f \notin (I \cup G)\} : \delta(I,\pi) \models \varphi} 1\right) - 0 \\ &= |\{\varphi \in I \cup \{\neg f \mid f \notin (I \cup G)\} \mid \delta(I, \pi) \models \varphi\}| \\ &= |\{\varphi \mid \overline{\varphi} \in \mathsf{PSE}(\mathcal{P}) \setminus \mathsf{SideEffect}(\mathcal{P}, \pi)\}| \\ &= |\mathsf{PSE}(\mathcal{P}) \setminus \mathsf{SideEffect}(\mathcal{P}, \pi)| \end{aligned}$$

Selecting π to maximize the cardinality of $\mathsf{PSE}(\mathcal{P}) \setminus \mathsf{SideEffect}(\mathcal{P}, \pi)$ minimizes the cardinality of $\mathsf{SideEffect}(\mathcal{P}, \pi)$, and vice versa, which gives us the desired result.

A.3 Proof of Proposition 2

As in Definition 22, let us say that the pairs that have non-zero probability under P_P are $\langle G_1, \pi_1\rangle, \ldots, \langle G_m, \pi_m\rangle$.

For $\vec{a}$ to be a plan-preserving plan for $\mathcal{P}$ means that it is a plan (so it makes G true) that maximizes the probability

$$\mathsf{P}_\mathsf{P}(\{\langle G', \pi'\rangle \mid \delta(\delta(I, \vec{a}), \pi') \models G'\}).$$

By Corollary 1 that is equivalent to maximizing

$$\mathsf{P}_\mathsf{P}(\{\langle G', \pi'\rangle \mid \delta(I, \vec{a}) \models \mathcal{R}^*(G', \pi')\}).$$

Since only $\langle G_1, \pi_1\rangle, \ldots, \langle G_m, \pi_m\rangle$ have non-zero probability under P_P, that is equivalent to maximizing

$$\sum_{j \in \{1,\ldots,m\} : \delta(I,\vec{a}) \models \mathcal{R}^*(G_j,\pi_j)} \mathsf{P}_\mathsf{P}(\langle G_j, \pi_j\rangle).$$

On the other hand, for $\vec{a} \in A_1^*$ to be an optimal plan for the plan-preserving compilation $\tilde{\mathcal{P}}$ means that it makes G true while maximizing the utility

$$\begin{aligned} u(\vec{a}) &= \left(\sum_{\varphi \in \mathcal{L}(F): \delta(I,\vec{a}) \models \varphi} u(\varphi) \right) - \sum_{i=1}^{k} c(a_i) \\ &= \left(\sum_{\substack{\varphi \in \mathcal{L}(F): \\ \delta(I,\vec{a}) \models \varphi}} \left(\sum_{\substack{j \in \{1,\dots,m\}: \\ \varphi = \mathsf{Con}(\mathcal{R}^*(G_j,\pi_j))}} \mathsf{P_P}(\langle G_j, \pi_j \rangle) \right) \right) - 0 \\ &= \sum_{j \in \{1,\dots,m\}: \delta(I,\vec{a}) \models \mathcal{R}^*(G_j,\pi_j)} \mathsf{P_P}(\langle G_j, \pi_j \rangle) \end{aligned}$$

So, what's being maximized by the plan-preserving plan for $\mathcal{P}$ and the optimal plan for $\tilde{\mathcal{P}}$ are the same.

A.4 Proof of Proposition 3

We'll write the pairs that have non-zero probability under $\mathsf{P_G}$ as $\langle G_1, i_1 \rangle, \dots, \langle G_m, i_m \rangle$.

Suppose that $\vec{a} \in A_1^*$ is any (not necessarily goal-preserving) plan for $\mathcal{P}$. Let us say that the *goal-preservation score* of $\vec{a}$ is the probability

$$\mathsf{P_G}(\{\langle G', i \rangle \mid \exists \vec{\pi} \in A_i^* \text{ s.t. } \delta(\delta(I,\vec{a}),\vec{\pi}) \models G'\}).$$

(That value is what a goal-preserving plan would maximize.) Because only $\langle G_1, i_1 \rangle, \dots, \langle G_m, i_m \rangle$ have non-zero probability, we can equivalently write the goal-preservation score as

$$\sum_{\substack{j \in \{1,\dots,m\} : \\ \exists \pi_j \in A_{i_j}^* \text{ s.t. } \delta(\delta(I,\vec{a}),\pi_j) \models G_j}} \mathsf{P_G}(\langle G_j, i_j \rangle)$$

Note also that the utility of a (not necessarily optimal) plan $\vec{b} \in \tilde{A}^*$ for the goal-preserving compilation $\tilde{\mathcal{P}}$ can be shown to be

$$u(\vec{a}) = \sum_{\substack{j \in \{1,\dots,m\} : \\ \delta(\tilde{I},\vec{b}) \models \{f_j \mid f \in G_j\}}} \mathsf{P_G}(\langle G_j, i_j \rangle)$$

The structure of the rest of this proof is as follows. We will show that any plan $\vec{a} = a_1, \dots, a_k \in A_1^*$ for $\mathcal{P}$ can be transformed into a plan for $\tilde{\mathcal{P}}$ that starts with the corresponding actions $\tilde{a}_1, \dots, \tilde{a}_k$ (and has no further actions from $\tilde{A}_{\mathsf{plan}}$), and whose utility is equal to the goal-preservation score of $\vec{a}$. Furthermore, any plan for $\tilde{\mathcal{P}}$ is of the form $\vec{\tilde{a}}, \vec{b}$, where $\vec{\tilde{a}} \in \tilde{A}_{\mathsf{plan}}^*$ and can be written as $\tilde{a}_1, \dots, \tilde{a}_k$ (for some k), and where $\vec{b} \in (\tilde{A}_{\mathsf{future}} \cup \tilde{A}_{\mathsf{advance}})^*$. From $\vec{\tilde{a}}$ can be extracted corresponding actions $a_1, \dots, a_k \in A_1^*$, which we will show form a plan for $\mathcal{P}$ whose goal-preservation score is at least as great as the utility of $\vec{\tilde{a}}, \vec{b}$ w.r.t. $\tilde{\mathcal{P}}$. From those things it will follow that the maximum possible goal-preservation score of a plan for $\mathcal{P}$ (i.e., that achieved by a goal-preserving plan) is equal to the maximum possible utility of a plan for $\tilde{\mathcal{P}}$ (i.e., that achieved by an optimal plan), and so we will get the result claimed in the proposition.

So, suppose that $\vec{a} = a_1, \dots, a_k \in A_1^*$ is a plan for $\mathcal{P}$ with a goal-preservation score of x. That means there exists a set $M \subseteq \{1, \dots, m\}$ such that

$$x = \sum_{j \in M} \mathsf{P_G}(\langle G_j, i_j \rangle),$$

and for each $j \in M$ (and no $j \in (\{1, \dots, m\} \setminus M)$) there exists $\pi_j \in A_{i_j}^*$ such that $\delta(\delta(I,\vec{a}),\pi_j) \models G_j$.

Now consider executing the following actions in $\tilde{A}^*$, starting from $\tilde{I}$:

- first, execute actions $\tilde{a}_1, \dots, \tilde{a}_k \in \tilde{A}_{\mathsf{plan}}^*$ (i.e., the corresponding actions to $a_1, \dots, a_k$)
- for each $j \in M$ (in increasing order):
  - execute each action of the form $switch_\ell$ (for any ℓ) that is executable, until $switch_j$ is executed
  - for each action $a \in \pi_j$ in order, execute the corresponding action $\tilde{a}^j \in \tilde{A}_{\mathsf{future}}$ (recall that π_j is the plan that achieved G_j from $\delta(I,\vec{a})$)

It's straightforward to verify that each of these actions will be executable, and the resulting state will satisfy the hard goal $\tilde{G} = \{f_0 \mid f \in G\}$ (that will be achieved by the actions from $\tilde{A}_{\mathsf{plan}}^*$, and then not changed), as well as $\{f_j \mid f \in G_j\}$ for each $j \in M$ (and no $j \in (\{1, \dots, m\} \setminus M)$). Therefore, the utility of the overall plan can be seen to be x.

To go in the other direction, consider any plan $\vec{b}$ for the goal-preserving compilation that has utility x. That means there exists a set $M \subseteq \{1, \dots, m\}$ such that

$$x = \sum_{j \in M} \mathsf{P_G}(\langle G_j, i_j \rangle),$$

and for each $j \in M$ (and no $j \in (\{1, \dots, m\} \setminus M)$) we have $\delta(\tilde{I},\vec{b}) \models \{f_j \mid f \in G_j\}$. Because of the preconditions of actions in $\tilde{A}$, it can be seen that $\vec{b}$ must be describable as follows:

- $\vec{b}$ starts with some actions $\tilde{a}_1, \dots, \tilde{a}_k \in \tilde{A}_{\mathsf{plan}}^*$ (for some k, possibly 0)
- then for each $j \in M$ (in increasing order)
  - there may be some actions from $\tilde{A}_{\mathsf{advance}}$
  - there is a (possibly empty) sequence of actions $\tilde{\pi}_j \in \tilde{A}_{\mathsf{future}}^*$, each action in $\tilde{\pi}_j$ being of the form $\tilde{a}^j$ (corresponding to some action $a \in A_{i_j}$)
- there are possibly some further actions in $\tilde{A}_{\mathsf{advance}}$ (though these could be omitted without reducing utility)

It can be shown that the action sequence $a_1, \dots, a_k \in A^*$ (corresponding to $\tilde{a}_1, \dots, \tilde{a}_k \in \tilde{A}_{\mathsf{plan}}^*$) will achieve G in the problem $\mathcal{P}$, and furthermore, from the resulting state, each G_j for $j \in M$ can be achieved by the plan $\pi_j \in A_{i_j}$ consisting of the corresponding actions to those in $\tilde{\pi}_j$. Therefore, the goal-preservation score of $a_1, \dots, a_k$ will be at least x.