Planning to Avoid Side Effects (Technical Appendix)

TECHNICAL REPORT CSRG-641, Department of Computer Science, University of Toronto

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

Introduction

This technical report provides additional information for the AAAI 2022 paper "Planning to Avoid Side Effects." Appendix A presents proofs of Theorems 2, 3, and 4, while Appendix B gives further details on the planning domains used in the experiments. See https://github.com/tqk/side-effects-planner for the code for the experiments, which was written in Python and heavily relies on the Tarski library (https://github.com/aig-upf/tarski).

A Proofs

This section contains proofs about the correctness of the fluent-, policy-, and goal-preserving compilations. First we have some background material.

A.1 Preliminaries

For uniformity with PSE scores and GSE scores, let us define *FSE scores*.

Definition 23 (FSE score)**.** The FSE score of a plan is the number of FSEs it has.

The following lemma will be useful in proving that planning problems are x-equivalent.

Lemma 1. *Let x be one of {fluent, policy, goal}. Suppose we have a STRIPS problem $\mathcal{P} = \langle F, I, \bigcup_{i=1}^{n} A_i, G\rangle$ and, if $x \in$ {policy, goal}, an appropriate set H and weight function w for specifying x side effect scores. Suppose that we have a planning problem $\mathcal{P}' = \langle F', I', A', G', c\rangle$ (with action costs c) and there is a set $A'_1 = \{a' : a \in A_1\} \subseteq A'$ such that:*

A. For any plan $\pi' \in A'^*$ for $\mathcal{P}'$, if the longest prefix of π' from A'^*_1 is $a'_1, \ldots, a'_k$, then $a_1, \ldots, a_k \in A^*_1$ is a plan for $\mathcal{P}$ with an x side effect score at most the cost of π'.

B. If $a_1, \ldots, a_k \in A^*_1$ is a plan for $\mathcal{P}$, then there is a plan for $\mathcal{P}'$ whose longest prefix of actions from A'^*_1 is $a'_1, \ldots, a'_k$, and with a cost at most the x side effect score of π.

Then $\mathcal{P}$ and $\mathcal{P}'$ are x-equivalent.

Proof. We first show two symmetrical properties:

- If a plan $\pi' \in A'^*$ for $\mathcal{P}'$ is cost-optimal – say with cost c_1 – then by (A) there is a plan $\pi \in A^*_1$ for $\mathcal{P}$ with an x side effect score of at most c_1, let's say c_2. By (B), there is a plan $\pi'' \in A'^*$ for $\mathcal{P}'$ with cost of at most c_2, let's say c_3. Since π' was cost-optimal, it follows that $c_1 = c_2 = c_3$. So the cost of π' is equal to the x side effect score of π.
- The other direction is symmetrical. If a plan $\pi \in A^*_1$ for $\mathcal{P}$ is x-preserving – say with an x side effect score of c_1 – then by (B) there is a plan $\pi' \in A'^*_1$ for $\mathcal{P}'$ that has cost $c_2 \leq c_1$. Then by (A) there is a plan $\pi \in A^*_1$ for $\mathcal{P}$ with an x side effect score of $c_3 \leq c_2$. Since π is x-preserving, $c_1 = c_2 = c_3$.

It follows that the optimal cost of a plan for $\mathcal{P}'$ is equal to the best possible x side effect score for a plan for $\mathcal{P}$, from which the result follows. ☐

Finally, we will go through some things that will be needed for the policy-preserving proof.

First, the following will be useful:

Corollary 1. *Given a STRIPS 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 $\langle F, s, A, \varphi\rangle$. ☐

Let us also prove the following property we claimed in the paper:

Proposition 1. *Given a STRIPS problem $\mathcal{P} = \langle F, I, A, G\rangle$, a goal-plan pair $h = \langle \hat{G}, \hat{\pi}\rangle$ (such that $\delta(I, \hat{\pi}) \models \hat{G}$) where $\hat{\pi} = a_1, \ldots, a_m$, and the policy ρ derived from $\hat{\pi}$ (and $\hat{G}$), we have for any state $s \in 2^F$ that* $\mathsf{achievable}(\hat{G}, \rho, s)$ *just in case $s \models \mathcal{R}^k(h)$ for some k.*

Proof. If $s \not\models \mathcal{R}^k(h)$ for any k, then $s \not\models \hat{G}$ and $\rho(s)$ is undefined, so we have $\mathsf{unachievable}(\hat{G}, \rho, s)$. The more interesting thing to show is that if $s \models \mathcal{R}^k(h)$ for some k, then $\mathsf{achievable}(\hat{G}, \rho, s)$

We can prove the result by (strong) induction on k.

Base case If $s \models \mathcal{R}^0(h)$, then $\hat{G}$ is already true and so ρ can trivially achieve it.

Inductive step The inductive hypothesis is that (for any state s) if $s \models \mathcal{R}^j(h)$ for some $j \leq k$, then $\mathsf{achievable}(\hat{G}, \rho, s)$.

Now suppose $s \models \mathcal{R}^{k+1}(h)$. If we also have that $s \models \mathcal{R}^j(h)$ for some $j \leq k$, then we have $\mathsf{achievable}(\hat{G}, \rho, s)$ and are done. Otherwise, by the construction of ρ, we have that $\rho(s) = a_{m-(k+1)+1}$. Let $s' = \delta(s, \rho(s))$. Since $s \models \mathcal{R}^{k+1}(h)$, by Corollary 1 we have that $\delta(s, a_{m-(k+1)+1}, \ldots, a_m) \models \hat{G}$, which can be restated as $\delta(s', a_{m-k+1}, \ldots, a_m) \models \hat{G}$. By Corollary 1 again, we then get that $s' \models \mathcal{R}^k(h)$. Then by the inductive hypothesis we have $\mathsf{achievable}(\hat{G}, \rho, s')$, and since $s' = \rho(s)$, we also have $\mathsf{achievable}(\hat{G}, \rho, s)$. ☐

A.2 Proof of Theorem 2

Suppose we have a STRIPS problem $\mathcal{P} = \langle F, I, A_1, G\rangle$, and let $\mathcal{P}' = \langle F', I, A', G', c\rangle$ be its fluent-preserving compilation. We want to show that $\mathcal{P}$ and $\mathcal{P}'$ are fluent-equivalent. Below we establish two properties that let us apply Lemma 1.

1. Consider any (not necessarily cost-optimal) plan $\pi' \in A'^*$ for $\mathcal{P}'$, where the longest prefix of π' from $A_1'^*$ is $a_1', \ldots, a_k'$. Note that after a_k', the next action must be end (so $a_1', \ldots, a_k'$ must achieve G), and then there must occur (possibly multiple instances of) exactly one of $\checkmark_f$ and $\times_f$ for each fluent $f \in F \setminus G$ (in order to make $noted_f$ true, as required by the goal). Furthermore, the number of fluents f for which $\times_f$ (with cost 1) appears in π' instead of $\checkmark_f$ (with cost 0) is equal to the number of fluents whose values have been changed by $a_1', \ldots, a_k'$ from the initial state. Furthermore, observe that each $a_i \in A_1$ changes the same fluents as $a_i' \in A_1'$. Therefore, not only does $a_1, \ldots, a_k$ achieve G in $\mathcal{P}$ (and so is a plan for $\mathcal{P}$), the cost of π' must be equal to the number of FSEs of $a_1, \ldots, a_k$.
2. Now consider any (not necessarily fluent-preserving) plan $\pi = a_1, \ldots, a_k \in A_1^*$ for $\mathcal{P}$. It's easy to see that can be converted into a plan for $\mathcal{P}'$, by starting with the actions $a_1', \ldots, a_k', end$, and then for each $f \in F \setminus G$, either adding $\checkmark_f$ or $\times_f$ depending on which has its precondition satisfied. Because of the precondition of $\times_f$, the cost of that plan will be equal to the number of FSEs of π.

The result then follows from Lemma 1. □

A.3 Proof of Theorem 3

Suppose we have a STRIPS planning problem $\mathcal{P} = \langle F, I, \bigcup_{i=1}^n A_i, G\rangle$, a finite set H of pairs $\langle \hat{G}, \hat{\pi}\rangle$ where $\hat{G} \subseteq F$ is a goal and $\hat{\pi} \in A_i^*$ (s.t. $\delta(I, \hat{\pi}) \models \hat{G}$), and a weight function $w : H \to \mathbb{R}$. We want to show that the policy-preserving compilation $\mathcal{P}'$ is policy-equivalent to $\mathcal{P}$.

Below we establish two properties that let us apply Lemma 1.

1. Consider any (not necessarily cost-optimal) plan $\pi' \in A'^*$ for $\mathcal{P}'$, where the longest prefix of π' from $A_1'^*$ is $a_1', \ldots, a_\ell'$. Note that after a_ℓ', the next action must be end (so $a_1', \ldots, a_\ell'$ must achieve G), and then there must occur exactly one of $\checkmark_h^k$ (for some k) and $\times_h$ for each fluent $h \in H$ (in order to make $noted_h$ true, as required by the goal). The cost of π' is equal to the sum of the costs of the $\times_h$ actions in it (and the cost of $\times_h$ is $w(h)$, the weight of h).

   For each action $\checkmark_h^k$ (let us say that $h = \langle \hat{G}, \hat{\pi}\rangle$) that is part of π', its precondition requires $\mathcal{R}^k(h)$, the regression of $\hat{G}$ through the last k actions in $\hat{\pi}$. It's easy to see if $\mathcal{R}^k(h)$ holds when $\checkmark_h^k$ is executed, then it must have held immediately after $a_1', \ldots, a_\ell'$ (since the end action and following actions cannot change fluents from F). Observe that each $a_i \in A_1$ changes the same fluents as $a_i' \in A_1'$. Therefore, not only does $a_1, \ldots, a_\ell$ achieve G in $\mathcal{P}$ (and so is a plan for $\mathcal{P}$), but it also achieves $\mathcal{R}^k(h)$ if $\checkmark_h^k$ is in π'. When that's the case, the policy derived from $\hat{\pi}$ can achieve $\hat{G}$ from $\delta(I, a_1, \ldots, a_\ell)$ (by Proposition 1).

   Therefore, if $h = \langle \hat{G}, \hat{\pi}\rangle \in H$ is such that the policy derived from $\hat{\pi}$ can *not* achieve $\hat{G}$ from $\delta(I, a_1, \ldots, a_\ell)$, it must be the case that $\times_h$ appears in π'. So the PSE score of $a_1, \ldots, a_\ell$ is at most the cost of π'.
2. Consider any (not necessarily policy-preserving) plan $\pi = a_1, \ldots, a_\ell \in A_1^*$ for $\mathcal{P}$. It's easy to see that π can be converted into a plan π' for $\mathcal{P}'$, by starting with $a_1', \ldots, a_\ell', end$, and then for each $h = \langle \hat{G}, \hat{\pi}\rangle \in H$ including one action as described below:
   - If the precondition of $\checkmark_h^k$ holds for some k, include $\checkmark_h^k$.
   - Otherwise, include $\times_h$. Note that for this case to apply, it must be that the policy derived from $\hat{\pi}$ cannot achieve $\hat{G}$ from $\delta(I, a_1, \ldots, a_\ell)$ in the original problem.

   So π' will include the action $\times_h$ just in case π has a PSE w.r.t. the policy derived from $\hat{\pi}$ and $\hat{G}$. So the cost of π' will be equal to the PSE score of π.

The result then follows from Lemma 1. □

A.4 Proof of Theorem 4

Suppose we have a STRIPS problem $\mathcal{P} = \langle F, I, \bigcup_{i=1}^n A_i, G\rangle$, a finite set H of pairs $\langle \hat{G}, i\rangle$ where $\hat{G} \subseteq F$ is a goal and i an agent (such that $\mathsf{achievable}(\hat{G}, i, I)$), and a weight function $w : H \to \mathbb{R}$. We want to prove that the goal-preserving compilation $\mathcal{P}'$ is goal-equivalent to $\mathcal{P}$

Below we establish two properties that let us apply Lemma 1.

1. Consider any (not necessarily cost-optimal) plan $\pi' \in A'^*$ for $\mathcal{P}$', where the longest prefix of π' from $A_1'^*$ is $a_1', \ldots, a_\ell'$. Note that after a_ℓ', the next action must be $clone$ (which requires that $a_1', \ldots, a_\ell'$ must achieve G). After $clone$, all the $acting_i$ fluents are false. It can be seen that the preconditions of actions (and the fact that the goal requires $noted_h$ for each $h \in H$) force the remainder of the plan to have this structure:
   - For each $h \in H$ (in any order)
     - First, $reset_i$ occurs.
     - 0 or more actions each from A_i' occurs (let us name this sequence of actions from $A_i'^*$ as π_h').
     - One of $\times_h$ and $\checkmark_h$ occurs.
   - Possibly additional actions occur afterwards, but no more $\times_h$ or $\checkmark_h$ actions can be executed (since $noted_h$ is already true for each h).

   The use of the $clone$ and $reset_i$ actions ensure that when π_h' is executed, the state of the fluents from F is the same as immediately after $a_1', \ldots, a_\ell'$. Furthermore, for each agent i, each action $a \in A_i^*$ changes the same fluents as $a' \in A_i'^*$. Therefore, it can be shown that for each

$h = \langle \hat{G}, i \rangle \in H$ such that $\checkmark_h$ appears in π', it must be the case that for the original problem $\mathcal{P}$,

$$\delta(\delta(I, a_1, \ldots, a_\ell), \pi_h) \models \hat{G}$$

where $\pi_h \in A_i^*$ consists of the "unprimed" actions corresponding to those in π'_h. Hence when $\checkmark_h$ appears in π', we see that $a_1, \ldots, a_\ell$ does not have a GSE on agent i w.r.t. goal $\hat{G}$. So for $a_1 \ldots, a_\ell$ to have a GSE on agent i w.r.t. goal $\hat{G}$ requires that $\times_h$ appear in π'.

We can conclude that $a_1, \ldots, a_\ell \in A_1^*$ is a plan for $\mathcal{P}$ whose GSE score is at most the cost of π'.

2. Consider any (not necessarily goal-preserving) plan $\pi = a_1, \ldots, a_\ell \in A_1^*$ for $\mathcal{P}$. We can convert π into a plan π' for $\mathcal{P}'$, by starting with $a'_1, \ldots, a'_\ell, clone$, and then for each $h = \langle \hat{G}, i \rangle \in H$ (in any order) appending the sequence of actions described below:
   - The first action appended is $reset_i$.
   - If $\mathsf{unachievable}(\hat{G}, i, \delta(I, \pi))$, the only additional action appended is $\times_h$.
   - Otherwise, there exists a plan $\pi_h \in A_i^*$ such that $\delta(\delta(I, \pi), \pi_h) \models \hat{G}$, and what's appended is the action sequence $\pi'_h \in A'^*_i$ consisting of "primed" versions of the actions in π_h, followed by $\checkmark_h$.

   It can be shown that that all the actions in π' will be executable, and the resulting state will satisfy G'. Furthermore, the cost of π' is equal to the sum of the weights associated with the pairs $h \in H$ such that $\times_h$ appears in π', which are precisely those pairs corresponding to GSEs of π. So the cost of π' is equal to the GSE score of π.

The result then follows from Lemma 1. □

B Domains

This section contains further details on the planning domains used in the experiments.

Aside from the Canadian wildlife domain described in the main paper, our experiments used adaptions of the standard IPC (International Planning Competition) domains *storage*,[4] *zenotravel*,[5] and *floortile*.[6] The PDDL (Planning Domain Definition Langauge) code for our versions can be found at https://github.com/tqk/side-effects-planner (the original versions that we adapted are available at http://planning.domains).

Below we quote from the description of each domain that can be found on http://planning.domains:

Zenotravel *The zenotravel domain involves transporting people around in planes, using different modes of movement: fast and slow. ...*

Floortile *A set of robots use different colors to paint patterns in floor tiles. The robots can move around the floor tiles in four directions (up, down, left and right). Robots paint with one color at a time, but can change their spray guns to any available color. However, robots can only paint the tile that is in front (up) and behind (down) them, and once a tile has been painted no robot can stand on it. ...*

Storage *Moving and storing crates of goods by hoists from containers to depots with spatial maps.*

For each domain we modified it to have separate agents:

- In storage the agents are the hoists.
- In zenotravel the agents are the aircraft.
- In floortile the agents are the painting robots.

We needed a way to mark an action as being from A_i, i.e. as being an action of the ith agent, in the PDDL files. We did so without changing the PDDL syntax by introducing an $acting_i$ fluent that we made a precondition of each action by agent i (for each i). We specified in the initial state that only the acting agent's $acting_i$ fluent was true. This notational convention means that if existing planners are applied to the PDDL files, the plans found will be ones for the acting agent (the other agent's actions will never be executable). Note that this means that our implementation of the goal-preserving compilation did not have to introduce its own $acting_i$ fluents but could use the existing ones.

Additionally, some of the typing for objects was modified (the original zenotravel did not have types), and action costs were removed from floortile.

[4]Introduced in IPC5: https://lpg.unibs.it/ipc-5/domains.html

[5]Introduced in IPC 2002: https://ipc02.icaps-conference.org/

[6]Introduced in IPC 2011: http://www.plg.inf.uc3m.es/ipc2011-deterministic/DomainsSequential.html