Computing Plans that Signal Normative Compliance

Alban Grastien
The Australian National University
Canberra, Australia
Alban.Grastien@anu.edu.au

Claire Benn
The Australian National University
Canberra, Australia
Claire.Benn@anu.edu.au

Sylvie Thiébaux
The Australian National University
Canberra, Australia
Sylvie.Thiébaux@anu.edu.au

ABSTRACT

There has been increasing acceptance that agents must act in a way that is sensitive to ethical considerations. These considerations have been cashed out as constraints, such that some actions are permissible, while others are impermissible. In this paper, we claim that, in addition to only performing those actions that are permissible, agents should only perform those courses of action that are *unambiguously* permissible. By doing so they signal normative compliance: they communicate their understanding of, and commitment to abiding by, the normative constraints in play. Those courses of action (or plans) that succeed in signalling compliance in this sense, we term 'acceptable'. The problem this paper addresses is how to compute plans that signal compliance, that is, how to find plans that are acceptable as well as permissible. We do this by identifying those plans such that, were an observer to see only part of its execution, that observer would infer the plan enacted was permissible. This paper provides a formal definition of compliance signalling within the domain of AI planning, describes an algorithm for computing compliance signalling plans, provides preliminary experimental results and discusses possible improvements. The signalling of compliance is vital for communication, coordination and cooperation in situations where the agent is partially observed. It is equally vital, therefore, to solve the computational problem of finding those plans that signal compliance. This is what this paper does.

CCS CONCEPTS

• **Computing methodologies → Planning and scheduling**; **Search with partial observations**; **Philosophical/theoretical foundations of artificial intelligence**.

KEYWORDS

ethics; planning; communication; uncertainty; constraint; permissibility; complexity

ACM Reference Format:
Alban Grastien, Claire Benn, and Sylvie Thiébaux. 2021. Computing Plans that Signal Normative Compliance. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society (AIES '21), May 19–21, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3461702.3462607

AIES '21, May 19–21, 2021, Virtual Event, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8473-5/21/05…$15.00
https://doi.org/10.1145/3461702.3462607

1 INTRODUCTION

Robots ought to be designed such that they are subject to ethical constraints [1, 11, 24, 34]. These constraints are likely to be case and context specific. What is permissible for one robot in one circumstance might be impermissible for another or in a different circumstance. Nevertheless, even if a robot is programmed to be sensitive to these ethical considerations, they are likely to interact with and be observed by humans who are unsure about whether or not the robot knows about the relevant normative constraints and is planning to abide by them. Against this background of general uncertainty, the robot's behaviour may well be morally ambiguous. Let's spell this out more slowly. The human observer is likely to only be able to observe part of the robot's behaviour (for example, when the robot engages in behaviour that is not easily directly observable; when the observer only observes the robot at certain time points; or when part of the plan is in the future and thus cannot be observed yet). This partially observed behaviour may be compatible with both permissible and impermissible courses of action. As such, the observer may have continued uncertainty about whether, in this instance, the robot in question has enacted or is planning on enacting a plan that contains impermissible acts. Thus, the robot's observed behaviour is morally ambiguous from the point of view of the observer.

Take the example of an empty self-driving car. Suppose that a pedestrian steps out on a pedestrian crossing and sees the car coming. The car's intended plan—to stop before hitting the pedestrian or to carry on thereby hitting the pedestrian—is not observable by the pedestrian. If the car plans to stop mere inches from the pedestrian, its course of action will (until the very last second) appear similar to the pedestrian to the plan to carry on across the crossing, hitting them. This course of action is therefore ambiguous: compatible with multiple plans. Moreover, it is *morally* ambiguous, because it is compatible with plans with different normative statuses: one is permissible (stopping mere inches from the pedestrian) and the other is impermissible (hitting them).

This moral ambiguity is a source of concern. When the observer has the power to intervene (for example, if the driver of a self-driving car was able to enact a manual override), it is likely to lead to inefficient or counterproductive interference with the working of the robot. Moreover, in situations where trust is important, the moral ambiguity of even permissible courses of actions are likely to fail to demonstrate trustworthiness or might in fact be detrimental to the relationship.

We argue, here and elsewhere [4, 5], that robot agents, just like human ones, ought to be cognizant of the *communicative* aspect of their behaviour and take seriously the imperative to *reassure* human observers by reducing moral ambiguity. They can do so, again just as humans do, by (what we call) signalling normative compliance: by choosing courses of action that are not only permissible but also are *unambiguously* permissible. Recall the example of the self-driving car: the plan to stop in *ample* time before the crossing signals the car's awareness of both the pedestrian and its commitment to the moral requirement not to hit the pedestrian. This plan is just as permissible as the plan to stop mere inches from the pedestrian; however, it is significantly less ambiguous.

The terminology we use throughout this paper is as follows. 'Permissible' refers to those courses of action that abide by the first-order normative constraints in play (for example, not to harm unnecessarily, not to cause catastrophic environmental damage, not to enter certain areas and so on). 'Acceptable' refers to those courses of action that are *unambiguously* permissible. How this is to be operationalised will be discussed in more detail below. Thus to 'signal normative compliance' is the selection of courses of action that are acceptable as well as permissible, such that the agent acts in a way that is normatively communicative. The problem this paper addresses is finding those plans that signal compliance, and finding them in a way that is sufficiently practical despite the high computational complexity of the problem.

The paper is organised as follows. We begin with a description of the running example used in the paper and in our experiments. We define the compliance signalling planning problem and reformulate it to reduce its complexity by adopting certain assumptions. In particular, we modify the formal definition of acceptability to be dependent on the cost-difference between the most cost-efficient impermissible plan and the most cost-efficient permissible plan, and establish the computational complexity of this reformulated problem.

We then propose an algorithm to solve this reformulated problem using an optimal classical planner as a subroutine. This algorithm computes increasingly expensive permissible plans and verifies whether these plans are acceptable. We then provide experimental results before discussing future and related work, and finally concluding.

2 EXAMPLE

Consider the simple logistic problem represented on Figure 1.[1] The goal is to drop a package currently in the truck at the target location (T), and drive the truck back to its current location (D). The truck can travel along the edges of the graph (the action $dr(x, y)$ moves the truck from x to y), and the cost of each leg is given in the figure.

The package is not permitted to enter the city (C) because it is hazardous.[2] However, the observer cannot see the path followed by the truck directly. Instead, the truck can send notifications from the Bx locations by performing action $no(Bx)$ (cost 0.1). Note that notifications can only be sent from these locations and not from the truck's starting point (D) or from the target location (T). Note also that these notifications record only the location, not the time at which the truck was there nor the direction from which it arrived or departed. The observer receives these notifications, in the order in which they were sent, but only after the truck has returned to its original location.

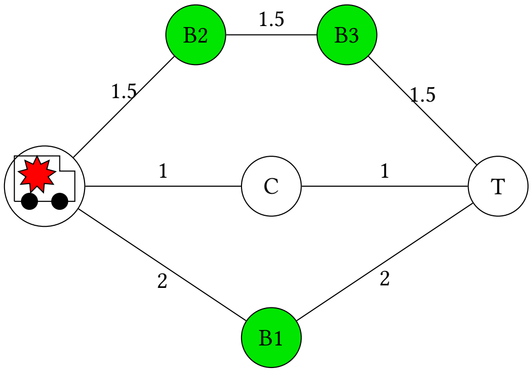

Figure 1: Logistic problem: the truck needs to drive from its starting location to T and back. It is impermissible for it to cross C before reaching T. In green cells, the truck can notify its locations. Cost of crossing an edge is indicated.

The cost-optimal plan would be to drive through the city and back ($\pi_1 = dr(D, C)$, $dr(C, T)$, $dr(T, C)$, $dr(C, D)$, cost = 4), but this plan is not permissible. This plan passes through none of the B locations and therefore affords no opportunity for notifications to be sent. The cost-optimal *permissible* plan is $\pi_2 = dr(D, B1)$, $dr(B1, T)$, $dr(T, C)$, $dr(C, D)$, cost = 6 . If the driver sends no notifications, this plan generates the same observation as π_1, namely no notification. Therefore, the observer will not be able to tell if the plan enacted is permissible or impermissible.

In order to prove that they did not enact plan π_1, the driver could send a notification when they cross B1: ($\pi_3 = dr(D, B1)$, $no(B1)$, $dr(B1, T)$, $dr(T, C)$, $dr(C, D)$, cost = 6.1).

The observer now knows that the truck travelled through B1, but on this basis alone does not know whether this was on the way to the target location or on the way back. This is because their observations are compatible with the following plan: $\pi_4 = dr(D, C)$, $dr(C, T)$, $dr(T, B1)$, $no(B1)$, $dr(B1, D)$, cost = 6.1. This is problematic because this latter plan is impermissible (as it involves carrying the package through the city).

A less ambiguous plan would be to cross and notify the locations B2 and B3 instead: $\pi_5 = dr(D, B2)$, $no(B2)$, $dr(B2, B3)$, $no(B3)$, $dr(B3, T)$, $dr(T, C)$, $dr(C, D)$, cost = 6.7. Because the observer sees $\sigma = [no(B2), no(B3)]$ (i.e., a notification from $B2$ followed by one from $B3$), they do not need to worry that the truck may have travelled through C with the package.

Indeed the cost-optimal impermissible plan consistent with σ is $\pi_6 = dr(D, C)$, $dr(C, D)$, $dr(D, B2)$, $no(B2)$, $dr(B2, B3)$, $no(B3)$, $dr(B3, T)$, $dr(T, C)$, $dr(C, D)$, cost = 8.7. While π_6 is consistent with the observations, it would be an irrational plan: why would the truck go to C and then come back? In this paper, our notion of rationality is conditioned on probabilities, which depend on the cost of the plan: π_5 is more rational than π_6 because it is significantly less expensive than π_6; the impermissible plans that explain σ have

[1] This example is inspired by some work on an industrial food chain that involves international partners over multiple jurisdictions with different legislations.

[2] Note that while this particular example is about safety in particular, our argument applies to any example with normative constraints. Issues such as safety are more easily agreed upon and thus we chose this example as it doesn't have any first-order normative disagreement to muddy the water, as the problem of signally compliance arises even when there is agreement by all parties on which actions are permissible and which are not. The ambiguity is about *compliance* with the constraints not about the content of the constraints.

a cost significantly higher than the cost of π_5, so π_5 is most likely the plan that was implemented. It is therefore acceptable.

3 COMPLIANCE SIGNALLING AS A PLANNING PROBLEM

In this section, we formulate the problem of finding an acceptable plan, in the sense of compliance signalling, within the framework of AI planning.

We start from classical planning, which is the problem of finding a course of action (a plan), enabling an agent to reach a given goal, starting from an initial state of the world. Actions have costs. These costs do not reflect the *moral* aspects of the action in question, and everything else being equal, a purely rational agent would (rationally) prefer the plan that has the smallest total cost, where 'rationality' is meant narrowly to capture a responsiveness to non-moral attributes and considerations.[3] An optimal classical planner, such as Fast Downward [15] is capable of returning such a least-cost plan.

Compliance signalling departs from classical planning by additionally imposing the constraint that the plan chosen by the agent must be acceptable. We start with the notion of a permissible plan: a plan that does not contain any action from a given impermissible set. Then, we turn to the notion of acceptability, which we capture probabilistically as follows: a plan is acceptable when the conditional probability mass, given the observation, of all impermissible plans that share this observation is below a given threshold.[4]

The prior probability distribution on plans reflects the fact the observer knows the agent is rational: everything else being equal, more costly plans are exponentially less likely.

This section formalises these ideas and is organised as follows. We start with some background on classical planning. Then we formalise the acceptability constraint, starting with its central aspects, namely the permissibility, observability and probability of plans. We then motivate making an assumption under which the constraint can be simplified to make acceptable plans easier to compute. Finally, we show, that, even with the simplified constraint, the generation of acceptable plans remains a much harder problem than classical planning.

3.1 Classical Planning

Classical planning is the problem of finding a sequence of actions enabling an agent to reach a given goal starting from an initial state of the world. The following assumptions are made: the initial state, the available actions, and the effects of these actions, are known to both the agent and the observer; moreover actions have deterministic effects. At this stage, we do not delve into how a planning instance is represented; more details are presented in Appendix A.

A *classical planning (problem) instance* is defined as a tuple $\mathbb{P} = \langle S, A, s_0, G, c\rangle$ where S is the set of states of the world, A is the set of available actions with $A(s)$ denoting the subset of actions applicable in state s, $s_0 \in S$ is the initial state, $G \subseteq S$ is the set of goal states, and $c : A \longrightarrow \mathbb{N}$ is the function specifying the cost of the actions. We write $s[a]$ for the state resulting from executing action $a \in A(s)$ in state s.

Let $\pi = \pi_1 \ldots \pi_n \in A^*$ be a sequence of actions whose ith element is π_i. π is a plan for the classical planning instance $\mathbb{P}$ iff all actions in π are applicable in turn from s_0, leading to successive states $s_0, s_1, \ldots, s_n$ where s_n is a goal state. That is, iff $s_n \in G$, and $\pi_i \in A(s_{i-1})$ and $s_i = s_{i-1}[\pi_i]$ for all $i \in \{1, \ldots, n\}$. The cost of the actions is additive, which means that the cost of a plan is the sum of each individual action cost: $c(\pi) = \sum_{i=1}^{n} c(\pi_i)$. A plan is optimal under some constraints if it satisfies the constraints and no other plan that also satisfy these constraints has a smaller cost.

3.2 The Compliance Signalling Constraint

As we explained before, the problem of compliance signalling is to find plans that are not only normatively permissible, but also visibly so, where visibility is defined with respect to a probability threshold. Formalising this constraint requires considering three main aspects: what makes a plan permissible/impermissible, what is the observation of a plan, what is the probability distribution of the plans from the point of view of an observer.

Permissibility. We assume that the set of actions is partitioned into permissible actions (denoted with $P \subseteq A$) and impermissible ones. A plan is then impermissible if it includes at least one impermissible action. We write $\Pi_{\mathrm{p}}(\mathbb{P})$ for the set of permissible plans of planning instance $\mathbb{P}$ and $\Pi_{\mathrm{i}}(\mathbb{P})$ for the impermissible ones.

This definition may sound restrictive. For instance, an action may be morally reprehensible only in some contexts so that one cannot classify the action as permissible or not in isolation. This is a difficult issue to solve. However, it can be overcome to a certain extent within the planning framework, by considering such actions to be differentiated on a more fine-grained description. Thus, instead of considering one action, on a coarse-grained description, to be permissible in some contexts but impermissible in another, we could consider there to exist two different actions that take into account the relevant contextual information, such that one of these actions is permissible and the other is impermissible. For instance, it may be permissible to pick up a fruit from a tree, but only if this tree is yours; in this case, we can distinguish between the action 'picking up a fruit from a tree that is yours' (which is permissible), and 'picking up a fruit from a tree that is not yours' (which is impermissible).

In fact, this framework can represent a large array of permissibility constraints. This includes constraints on the temporal occurrence of actions in the plan, such as those traditionally captured by temporal logic over finite traces [2, 9], e.g. 'it is permissible to pick up a fruit only after helping its owner'. It is well known that such constraints can be compiled away by redefining the states to capture the necessary information [29].[5] For the example above, it suffices to modify the states to record which owners you helped

[3] Of course, this is not to say that they would prefer the plan with the smallest total cost *all things considered*, as they may well take into account the moral assessment of the plan, which is not factored into the costs. In fact, in this paper, we assume that the agent is committed to respecting these normative aspects and to communicating that commitment to an uncertain observer.

[4] There are other possible forms that this constraint could take. Elsewhere we explore in detail some of the alternative ways of cashing out acceptability, including, for example, incorporating the observer's beliefs about the agent's preferences [5].

[5] For LTL, this however increases the size of the state space exponentially in the size of the formula in the worst case.

so far; then, the set of actions will include 'pick up a fruit from a tree whose owner you helped' (permissible, but applicable only if you indeed helped its owner) and 'pick up a fruit from a tree whose owner you haven't helped' (impermissible).

Some constraints do not refer only to the current plan, but also to other possible plans. It is a core part of some ethical views that certain normative features can only be determined in relation to the alternatives available. See, for example, [3, 13, 26, 33]. For instance, it may be permissible to pick up a fruit from a tree that is not yours as long as you are unable to pick up fruits from your own trees. Future work will investigate whether existing work on compiling away preferences in planning [18, 35] enables this type of constraint to be encoded in our framework.[6]

Observability. The observer only has partial observation of the plan executed by the agent. We assume that the set of actions is partitioned into the observable actions, whose set is denoted by O, and the unobservable ones. As with permissible actions, this may require us to redefine the actions and the states. The observation of a plan π is then the projection of the plan over the observable actions which is formally defined as follows.

Given an action a, and a plan π, we define $obs(\pi)$ as follows, where $a.L$ is the list obtained by prepending a to the list L.

$$obs(\pi) = \begin{cases} [] & \text{if } \pi = [] \text{ (empty plan)} \\ a.obs(\pi') & \text{if } \pi = a\pi' \text{ and } a \in O \\ obs(\pi') & \text{if } \pi = a\pi' \text{ and } a \notin O. \end{cases}$$

Thus, $obs(\pi)$ is the list that contains all those actions that compose π that are observable, in order (and thus, is empty if no action is observable). Take the plan π consisting of the sequence of actions π =*go(D,B2), no(B2), go(B2,B3), no(B3), go(B3,T)*. Since only the $no(\cdot)$ actions are observable, $obs(\pi) = [no(B2), no(B3)]$.

Two plans are indistinguishable, denoted by $\pi_1 \# \pi_2$, if they generate the same observation: $obs(\pi_1) = obs(\pi_2)$. Plans with a different length may generate the same observation. Note we make the assumption that the indistinguishability of observations is a transitive property, i.e., $\pi_1 \# \pi_2$ and $\pi_2 \# \pi_3$ implies $\pi_1 \# \pi_3$.

This is not an innocuous assumption as, in some other contexts, indistinguishability is not transitive. For example, something small and something very small might be indistinguishable, as could be something very small and something extremely small; however, this does not entail that something small and something extremely small are similarly indistinguishable. Nevertheless, we assume that transitivity holds for the indistinguishability of observations in this context.

An observation is denoted by the symbol σ. The set of plans that produce the observation σ (i.e., such that $obs(\pi) = \sigma$) is denoted Π^σ.

Probability. The last ingredient necessary to formalise the idea of signalling compliance is the way to estimate the probability of each plan given an observation. The assumption often made in plan recognition [27] is that, all other things being equal, the probability of a plan depends only on the cost of the plan. More specifically, given that the observer sees the observation σ when a plan is executed by the agent, the probability that this plan is π, noted $\Pr(\pi \mid \sigma)$ is

$$\Pr(\pi \mid \sigma) = \begin{cases} \alpha \times e^{-\beta \times c(\pi)} & \text{if } obs(\pi) = \sigma \\ 0 & \text{otherwise,} \end{cases}$$

where β is a constant that indicates how quickly the probability of a plan drops with its cost, and α is a normalisation factor that guarantees that the probabilities add up to 1. More specifically,

$$\alpha = \frac{1}{\Sigma_{\pi' \in \Pi^\sigma} \; e^{-\beta \times c(\pi')}}.$$

The probability of a plan being the one performed by the agent drops exponentially fast with its cost. In particular, if $c(\pi_1) - c(\pi_2) = c(\pi'_1) - c(\pi'_2)$, then

$$\frac{\Pr(\pi_1 \mid \sigma)}{\Pr(\pi_2 \mid \sigma)} = \frac{\Pr(\pi'_1 \mid \sigma)}{\Pr(\pi'_2 \mid \sigma)}.$$

In words, this means that in order to compare the probabilities of two plans, one just needs to look at the difference in cost between these two plans.

Acceptability. We now have all the elements necessary to formalise the compliance signalling constraint. We define a *compliance signalling planning (problem) instance* as a tuple $\mathcal{P} = \langle \mathbb{P}, P, O, \varepsilon \rangle$, where $\mathbb{P} = \langle S, A, s_0, G, c \rangle$ is a classical planning instance, $P \subseteq A$ is the set of permissible actions, $O \subseteq A$ is the set of observable actions, and ε is a probability threshold whose role will become clear below.

We now define the notion of an acceptable plan for $\mathcal{P}$. Let σ be the observation made by the observer. The probability that σ is the observation of an impermissible plan is the sum of the conditional probabilities of the impermissible plans that match the observation σ:

$$\Pr(\text{impermissible} \mid \sigma) = \Sigma_{\pi' \in \Pi_{\text{i}}(\mathbb{P})} \; \Pr(\pi' \mid \sigma).$$

A plan that produced observation σ is acceptable iff this probability is below the threshold ε.

Definition 1. *Let $\mathcal{P} = \langle \mathbb{P}, P, O, \epsilon \rangle$ be a compliance signalling planning instance. A plan π for the classical planning instance $\mathbb{P}$ is* acceptable *for $\mathcal{P}$ iff*

$$\Pr(\text{impermissible} \mid obs(\pi)) \leq \varepsilon. \tag{1}$$

Our goal is to compute the optimal compliance signalling plan, i.e., the permissible plan with minimal cost that is nevertheless acceptable.

3.3 Reformulating and Simplifying Acceptability

Deciding whether a plan π is acceptable is significantly more difficult than finding a classical plan which, itself, is already PSPACE-complete [6]. Indeed, it requires, in the worst case, to compute all paths that generate the same observation as π, and there could be infinitely many of those. Finding an acceptable plan is even harder.

Because of this complexity, we propose and justify a reformulation of the problem into a more amenable one which only requires considering two plans that generate the same observation, rather than infinitely many. Our reasoning is based on the following assumption about the *probability mass fraction* of the cost-optimal plans which we first define.

[6] One issue might be that these compilations affect the cost of actions. This could interact with other aspects of the framework presented here.

Given an observation σ and a plan $\pi \in \Pi^\sigma$, the probability mass fraction $k(\pi) \in [0, 1]$ is the ratio of the conditional probability of this plan, given the observation in question, over the conditional probability of all plans of the same normative class (permissible or impermissible), given that same observation:

$$k(\pi) = \frac{\Pr(\pi \mid \sigma)}{\Pr(\text{class}(\pi) \mid \sigma)}.$$

In other words, $k(\pi)$ indicates how much of the probability of its class can be "attributed" to the plan π. We are particularly interested in the optimal plans of each class. Given an observation σ, we write π_i^σ and π_p^σ for the plans that are optimal amongst Π_i^σ and Π_p^σ. Because of the definition of $\Pr(\pi \mid \sigma)$, the optimal plan will have the maximal contribution of its class.[7]

To simplify and reformulate the problem, we make the assumption that the probability mass fractions of the cost-optimal permissible plan and the cost-optimal impermissible plan are comparable.

Assumption 1. *The probability mass fraction of π_p^σ and π_i^σ are similar:*

$$k(\pi_\text{p}^\sigma)/k(\pi_\text{i}^\sigma) \simeq 1.$$

It is now possible to rewrite Equation 1 as follows:

$$\varepsilon \geq \frac{\Sigma_{\pi \in \Pi_\text{i}^\sigma(P)}\ e^{-\beta \times c(\pi)}}{\Sigma_{\pi \in \Pi_\text{i}^\sigma(P)}\ e^{-\beta \times c(\pi)} + \Sigma_{\pi \in \Pi_\text{p}^\sigma(P)}\ e^{-\beta \times c(\pi)}}$$

We can use the definition of the probability mass fraction to mention only the cost optimal plans.

$$\varepsilon \geq \frac{e^{-\beta \times c(\pi_\text{i}^\sigma)}/k(\pi_\text{i}^\sigma)}{e^{-\beta \times c(\pi_\text{i}^\sigma)}/k(\pi_\text{i}^\sigma) + e^{-\beta \times c(\pi_\text{p}^\sigma)}/k(\pi_\text{p}^\sigma)}$$

We can remove the ks as they are similar:

$$\varepsilon \geq \frac{e^{-\beta \times c(\pi_\text{i}^\sigma)}}{e^{-\beta \times c(\pi_\text{i}^\sigma)} + e^{-\beta \times c(\pi_\text{p}^\sigma)}}$$

$$e^{-\beta \times c(\pi_\text{p}^\sigma)} \geq e^{-\beta \times c(\pi_\text{i}^\sigma)} \frac{1 - \varepsilon}{\varepsilon}$$

$$-\beta c(\pi_\text{p}^\sigma) \geq -\beta c(\pi_\text{i}^\sigma) + \ln\left(\frac{1 - \varepsilon}{\varepsilon}\right)$$

$$c(\pi_\text{p}^\sigma) \leq c(\pi_\text{i}^\sigma) - \ln\left(\frac{1 - \varepsilon}{\varepsilon}\right)/\beta$$

We use the notation $\delta = \ln(\frac{(1-\varepsilon)}{\varepsilon})/\beta$. Then the plan π_p^σ is acceptable iff the following inequality holds:

$$c(\pi_\text{p}^\sigma) \leq c(\pi_\text{i}^\sigma) - \delta. \tag{2}$$

This equation states that an observer will be convinced that the plan is permissible if the most rational permissible explanation (i.e., the optimal permissible plan consistent with the observation) is less expensive than the most rational impermissible one. As was stated before, the ratio of the probabilities of the two plans, $\Pr(\pi_\text{p}^\sigma)/\Pr(\pi_\text{i}^\sigma)$, is a function of the difference of their cost.

We now summarise with our simplified definition of acceptability:

Definition 2. *Let $\mathcal{P} = \langle \mathbb{P}, P, O, \epsilon \rangle$ be a compliance signalling planning instance. A plan π for the classical planning instance $\mathbb{P}$ is* acceptable in the simplified sense *for $\mathcal{P}$ iff*

$$c(\pi_\text{p}^{obs(\pi)}) \leq c(\pi_\text{i}^{obs(\pi)}) - \delta$$

where $\pi_\text{p}^{obs(\pi)}$ (resp. $\pi_\text{i}^{obs(\pi)}$) is the cost-optimal permissible (resp. impermissible) plan satisfying the same observations as π, and $\delta = \ln(\frac{1-\epsilon}{\epsilon})/\beta$.

3.4 Complexity

Determining whether a classical planning instance has a solution is PSpace-complete [6]. We now show that even with our simplified notion of acceptability, determining whether a compliance signalling planning instance has an acceptable solution is substantially harder than classical planning.

Definition 3. *Let $\mathcal{P}$ be a compliance signalling planning instance. The* acceptability decision problem consists *in deciding whether there exists a plan π which is acceptable in the simplified sense for $\mathcal{P}$.*

Theorem 1. *The acceptability decision problem (Def 3) is ExpSpace-hard.*

The proof of complexity relies on a reduction from *conformant planning* to the acceptability decision problem. Since it is known [14] that conformant planning is ExpSpace-complete, we know that deciding whether there exists an acceptable plan is ExpSpace-hard.

The full reduction is given in Appendix B, but we give the intuition here. Conformant planning is similar to classical planning except that some actions have non-deterministic effects and that their outcome is unobservable. Therefore, a sequence of actions is a solution for a conformant planning problem iff 1) it is applicable regardless of the actions' outcomes and 2) its execution always reaches the goal. For instance, in the *Bomb in the Toilet* domain [32], a robot is supposed to dunk bombs into a toilet; executing this action may or may not clog the toilets (non-deterministic effect) and whether it does or not is unobservable; therefore, a valid solution requires the robot to flush the toilets after each dunk, just in case the toilet was clogged. The goal of the problem is generally represented by a condition on the final state, but for simplicity, and without loss of generality, we assume that the goal is to perform a specific action.

The conformant planning problem and the acceptability decision problem have a similar structure. Indeed, conformant planning requires finding a sequence of actions such that all outcomes of these actions successfully lead to the goal. In comparison, acceptable planning requires finding a sequence of observable actions such that all plans that match this sequence are permissible.

4 ALGORITHM & IMPLEMENTATION

Following the previous section, we propose a simple algorithm for signalling compliance, which builds on classical planning to return a cost-optimal permissible and acceptable plan. Our procedure is described in Algorithm 1. Importantly before getting into further details, we want to note that the procedure repeatedly calls for a classical planner to solve variants of the original classical planning instance $\mathbb{P}$ augmented with extra constraints (such as 'the plan should (not) be permissible', or 'the plan should (not) generate these

[7] If these plans do not exist, i.e., if there is no impermissible (resp. permissible) plan that match the observation σ, we consider that π_i^σ (resp. π_p^σ) is a dummy plan with infinite cost (and, therefore, 0 probability).

Algorithm 1 Signalling Compliance in Classical Planning

```
1:  input: $\mathbb{P} = \langle A, I, G, c \rangle$ planning problem
2:  input: $O \subseteq A$ observable actions
3:  input: $P \subseteq A$ permissible actions
4:  input: $\delta$ cost differential threshold
5:  $\mathcal{L}_{\text{forb}} := \emptyset$
6:  loop
7:      Compute a cost-minimal permissible plan $\pi_{\text{p}}$ solution to $\mathbb{P}$
        such that $obs(\pi_{\text{p}}) \notin \mathcal{L}_{\text{forb}}$.
8:      Compute a cost-minimal impermissible plan $\pi_{\text{i}}$ solution to $\mathbb{P}$
        such that $obs(\pi_{\text{i}}) = obs(\pi_{\text{p}})$.
9:      if $c(\pi_{\text{p}}) \leq c(\pi_{\text{i}}) - \delta$ then
10:         return $\pi_{\text{p}}$
11:     end if
12:     Add $obs(\pi_{\text{p}})$ to $\mathcal{L}_{\text{forb}}$
13: end loop
```

observations'). We explain how these constraints are incorporated into $\mathbb{P}$ in Appendix A.

Algorithm 1 searches for a cost-optimal permissible plan π_{p}, and then for a cost-optimal impermissible plan π_{i} that produces the same observation. If the cost of these plans satisfy the condition $c(\pi_{\text{p}}) \leq c(\pi_{\text{i}}) - \delta$, then π_{p} is an acceptable plan and is returned by the procedure. If, on the other hand, this condition is not satisfied, then π_{p} is not acceptable; furthermore, no plan π' that generates the same observation $obs(\pi_{\text{p}})$ is acceptable, since that plan would have a cost even higher:

$$c(\pi') \geq c(\pi_{\text{p}}) > c(\pi_{\text{i}}) - \delta.$$

We store all the observations $obs(\pi)$ of failed candidate plans π into the *forbidden observable language* $\mathcal{L}_{\text{forb}}$, and enforce that later candidate plans produce an observation that is not in $\mathcal{L}_{\text{forb}}$.

We illustrate this with the example of Section 2. Initially, the language $\mathcal{L}_{\text{forb}}$ is empty. The first permissible plan $\pi_2 = dr(D, B1)$, $dr(B1, T)$, $dr(T, C)$, $dr(C, D)$ is proved to be unacceptable. As this plan's observation is the empty one, $obs(\pi_1) = []$, and this observation is added to $\mathcal{L}_{\text{forb}}$ which now contains a single word $\{[]\}$. A second plan $\pi_3 = dr(D, B1)$, $dr(B1, T)$, $no(B1)$, $dr(T, C)$, $dr(C, D)$ is proved to be unacceptable. This plan's observation is $obs(\pi_3) = [no(B1)]$. It is added to $\mathcal{L}_{\text{forb}}$ which is now: $\{[], [no(B1)]\}$. We can see that $\mathcal{L}_{\text{forb}}$ grows incrementally until the optimal permissible plan whose observation is not in $\mathcal{L}_{\text{forb}}$ is acceptable.

Algorithm 1 requires the classical planner used in lines 7 and 8 to return an optimal plan. It also requires the classical planner to be able to determine when there is no plan (in particular if there is no impermissible plan that matches the observation); to simplify notations and without loss of generality, we assume that the classical planner returns a plan with infinite cost when it proves that there is none.

Lemma 1. *If all observable actions have a strictly positive cost and if there is at least one acceptable plan, then Algorithm 1 returns the cost-optimal acceptable plan.*

Proof sketch: by contradiction. Assume that Algorithm 1 is not able to return the cost optimal plan π_{p}^*. Then, either it returned another plan or it never terminates. In the first case, then either it returns an unacceptable plan (which is not possible since it only returns plans that pass the acceptability condition) or it returns a suboptimal plan π' (which is also impossible since the call to the classical planner on Line 7 would not return π' over π_{p}^*). If the procedure never terminates, then it implies that the classical planner generates an infinite sequence of plans $\pi_1, \pi_2, \ldots$ that all produce different observations. As a consequence, the length of these observations is unbounded (for any integer y, there exists an index j such that $|obs(\pi_j)| > y$). Furthermore, these plans all have a cost less than or equals to that of π_{p}^* since we assume that π_{p}^* is never produced. However, notice that the cost of all observable events of a plan is a lower bound for the cost of the plan: $c(\pi) \geq c(obs(\pi))$. Let $x > 0$ be the minimal cost of any observable action. Then any sequence of observable actions with at least $\lceil c(\pi_{\text{p}}^*)/x \rceil$ elements will have a cost higher than that of π_{p}^*. This implies that the procedure will eventually produce a plan π_j with a cost higher than the solution, which contradicts the assumption of optimality of classical planner.

The procedure of Algorithm 1 may not terminate if there is no acceptable plan. It seems that there is no upper bound on the length of the plan π_{p}^*.

5 EXPERIMENTS

In this section, we present an experimental evaluation of Algorithm 1, in which we study how well the algorithm scales.

We implemented Algorithm 1 in Java. This implementation is responsible for calling the classical planner, but also for generating the modified planning instances as explained in Appendix A, e.g., to specify that the plan should generate a specific observation. The classical planner used is Fast Downward [15] running A^* with the $h^{\max}$ heuristic.

We used instances of the logistics problem introduced in Section 2. Our instances are defined according to 4 parameters: the number w of alternate paths (for instance, $B2$–$B3$ represent a single path); the length d of these alternate paths (the path $B2$–$B3$ has length 2); the number p of packages to deliver (half of which are dangerous); the number t of trucks. A truck can only pickup one dangerous package at a time (but there is no limit on the non-dangerous ones). We choose a δ value of 2.

We make the following observations on these parameters. If $w = d = 1$, there is no acceptable plan as we illustrated in the example.

Increasing the number of trucks can make the problem harder because it increases the number of symmetries. If, for instance, our algorithm generates an unacceptable plan that involves one or several trucks, then it will generate all the possible permutations of trucks (as long as they lead to different observations) before moving on to other types of plans. On the other hand however, adding trucks can reduce the cost of the optimal plan (which makes it easier to find). Additional packages however only make the problem harder, i.e., they increase the length of the plans and the number of symmetries in the plans.

The results are presented in Table 1. We gave a time limit of 15 minutes per problem instance. The table shows the instance parameters (w, d, p, t), and the run-time and number of iterations of the algorithm.

Instance	time (s)	iter	Instance	time (s)	iter	Instance	time (s)	iter
(1,2,1,1)	1.044	6	(4,1,1,2)	2.445	12	(4,4,2,1)	5.686	20
(1,2,2,1)	1.047	6	(3,1,1,3)	2.984	13	(2,1,3,1)	6.247	24
(2,1,1,1)	1.070	6	(4,1,2,2)	3.115	12	(2,2,1,3)	7.009	21
(1,2,1,1)	1.148	6	(4,2,2,1)	3.133	15	(3,1,2,3)	7.458	13
(3,1,1,1)	1.187	7	(1,2,1,3)	3.150	12	(2,1,2,4)	12.52	12
(3,1,2,1)	1.216	7	(2,2,1,2)	3.245	15	(4,1,2,3)	14.12	16
(2,1,2,1)	1.395	6	(4,2,1,1)	3.467	15	(2,1,4,1)	14.45	24
(4,1,2,1)	1.437	8	(2,1,1,4)	3.470	12	(2,2,1,4)	14.90	27
(4,1,1,1)	1.442	8	(4,3,1,1)	3.531	16	(1,2,4,1)	15.47	22
(2,1,1,2)	1.531	8	(4,3,2,1)	3.858	16	(1,2,2,4)	17.46	15
(2,2,1,1)	1.655	9	(2,1,2,3)	4.103	10	(3,1,3,1)	19.06	51
(2,1,2,2)	1.724	8	(4,1,1,3)	4.338	16	(2,2,2,3)	21.73	21
(1,2,1,2)	1.740	9	(1,2,1,4)	4.660	15	(3,1,2,4)	37.56	16
(3,1,1,2)	1.924	10	(2,2,2,2)	4.792	15	(3,1,4,1)	80.59	51
(2,2,2,1)	1.998	9	(4,4,1,1)	5.090	20	(4,1,3,1)	88.86	100
(3,1,2,2)	2.250	10	(1,2,2,3)	5.277	12	(2,2,2,4)	134.0	27
(1,2,2,2)	2.314	9	(3,1,1,4)	5.358	16	(2,2,3,1)	148.4	117
(2,1,1,3)	2.415	10	(1,2,3,1)	5.494	22	(1,2,3,2)	666.7	124

Table 1: Experimental results showing the time to compute the optimal acceptable plan and the number of iterations of the algorithm.

We see clearly that the problem is much more difficult than classical planning. The largest instance that was solved, specifically $(w = 2, d = 2, p = 2, t = 4)$, only includes $290,521$ states (note however that this number of states increases significantly once the extra constraints are incorporated into the instance).

The difficulty lies mostly in the fact that the planner must be called repeatedly, as demonstrated by the correlation between the number of iterations and the total runtime. Remember that the number of calls to the classical planner is twice the number of iterations, since the first call is used to produce a permissible plan, and the second call used to decide whether the plan is acceptable.

Looking at the behaviour of Algorithm 1, there are multiple situations where it is too naïve. We mentioned above that the instances contain numerous symmetries, and we could use symmetry-breaking techniques. Even without that, one issue with our procedure is that it is not able to generalise from its results, as we show now.

Consider the example of Figure 1. We already mentioned that the optimal permissible (unacceptable) plan was $\pi_2 = dr(D, B1)$, $dr(B1, T)$, $dr(T, C)$, $dr(C, D)$, and that a first (unsuccessful) attempt to make it acceptable would be to insert a notification during the plan: $no(B1)$. An obvious follow-up to this attempt is to insert a second notification; this is also unacceptable since both notifications could have been added on the way back, and of course no amount of notifications $no(B1)$ will change that. Since we indicated that the cost of performing the notification is negligible, Algorithm 1 will consider all variants of π_2 with an increasing number of notifications before even considering other options. We would like to be able to learn and include the information that adding more notifications does not help improve the acceptability of the plan.

So the question we ask is: given a permissible plan and an indistinguishable impermissible plan, is it possible to analyse this pair in order to determine a minimal change required to make a plan acceptable? This is an open question, and it might be possible to answer it by looking at the causal relationship between the different actions, or through an analysis similar to the conflict detection in model based diagnosis [12, 28].

6 CONCLUSION, RELATED AND FUTURE WORK

Agents in all domains need to be sensitive to ethical considerations when selecting plans. Moreover, as we have claimed, they must also be sensitive to the *appearance* of those plans. In particular, they must be sensitive to the fact that those plans may appear morally ambiguous—that is, might appear impermissible, even when they are permissible—to observers able only to see part of them. Agents should therefore signal their normative compliance by selecting those plans that are not only permissible but also acceptable: that are unambiguously permissible.

The issue of compliance signalling is central to the design of ethically-aware AI systems, especially when these directly interact with humans. Recognising this fact demands that designers of such systems build them to act in ways that are not just permissible but also acceptable. In order for such signalling to become the norm in industry and in public expectation, we must determine how to find those plans that are acceptable. We have formalised the problem of synthesising compliance signaling plans, shown that it is computationally difficult (more difficult than classical planning), and proposed an iterative algorithm that finds increasingly costly candidate permissible plans using a classical planner, and verifies their acceptability, by comparing their cost with that of the cost-minimal impermissible plan sharing the same observation.

The importance of anticipating an observer's interpretation has been recognised in areas such as plan recognition [27], goal recognition design [17], model based diagnosis [16], legible motion [7, 10, 25], implicit cues [20] and trust [21, 30]. Indeed the relationship we use between the probability and cost of a plan is standard in plan recognition. These debates however rarely address explicitly the *normative* aspects of agent behaviour.

Recently Lindner, Mattmüller, and Nebel [22] started to investigate the problem of implementing planning systems that comply with ethical principles. They show that the complexity of *verifying* whether a plan is ethical varies considerably depending on the ethical principle chosen. For instance, deciding whether a plan is morally permissible according to deontology or utilitarianism is polynomial or PSPACE-complete, respectively. They do not however consider signalling compliance and how these plans might appear to an observer with partial knowledge of the plan. As we have shown, even considering a deontic notion of permissibility, deciding whether a plan is acceptable is EXPSPACE-hard.

Computationally, compliance signalling planning is related to the generation of legible or transparent plans, described by e.g. [7, 19, 23]. These works formalise the problem of generating plans that clearly convey (or conversely obfuscate) the goal of a planning agent to an external observer. The motivation is to cater for cooperative and adversarial scenarios. For instance, Kulkarni et al. [19] studies the generation of plans achieving the agent's goal such that there are at most j plans with the same observations that reach potentially confounding goals. The algorithms used in these works differ substantially from ours and explicitly manipulate belief states. For instance the powerful algorithm of [23], solves a goal POMDP (partially observable Markov decision process) whose transition probabilities reflect the evolution of the observer's belief state, as computed by a probabilistic goal recognition system. In contrast, our algorithm only requires an optimal classical planner.

As discussed in Section 5, there are many opportunities to make our implementation and algorithm more practical, including by breaking symmetries [31] and learning conflicts [12]. On the other hand, our algorithm makes a range of assumptions, which our future work agenda will attempt to weaken. This includes establishing the decidability and complexity of the original (non-simplified) acceptability notion we introduced. At the same time, we would like to consider richer scenarios in which multiple observers have different moral viewpoints and observation capabilities, as well as world models that potentially differ from that of the agent [8]. Determining an acceptable plan relative to *all* of those observers, or alternatively, determining *which* observer's perspective should be prioritised, adds further complexity to the problem. Finally, an important limit is that we consider an *a posteriori* setting, in which the entire observation of a plan is considered to determine its acceptability, whereas there is a need to generate plans that continually reassure the observer.

Communication and compliance is central in the normative domain. This work, and future work, on the *implementation* of communicating compliance in AI systems is vital if this key aspect of moral behaviour is to be realised and respected. It is not a simple problem. However, this paper provides an important first step.

A CONSTRAINTS ENCODING

In order to explain how we enforce the constraints on the plans that are required by Algorithm 1, we first need to briefly discuss the propositional STRIPS formalism used to model planning problems. Then, in order to incorporate the constraints used in Algorithm 1, we formulate each constraint as a test for membership of a language recognised by a deterministic finite automata (DFAs), and we finally compile those DFAs into the STRIPS problem description.

A.1 Propositional STRIPS

A propositional STRIPS model relies on a set of facts $\mathcal{F}$ that are used to describe the world. A fact is either true or false in a state, and a state is described precisely as the set of facts that true in it: $s \subseteq \mathcal{F}$. An action a is then defined as a triple $\langle \text{pre}, \text{eff}^+, \text{eff}^- \rangle$ where pre is the *precondition* of the action, eff^+ its *positive effects*, and eff^- its *negative effects*. All three elements are subsets of $\mathcal{F}$. An action is applicable in a state s if its precondition is satisfied, i.e., $\text{pre} \subseteq s$. The negative effects represent the facts that are made false by the action, the positive effects those that are made true. Hence the state s' reached by applying the action in state s is described by $(s \setminus \text{eff}^-) \cup \text{eff}^+$.

A.2 Deterministic Finite Automata

A deterministic finite automaton (DFA) is a tuple $\mathcal{A} = \langle Q, \Sigma, T, q_0, F \rangle$ where Q is a set of states, Σ is a set of labels, $T : Q \times \Sigma \longrightarrow Q$ is a transition function, $q_0 \in Q$ is an initial state, and $F \subseteq Q$ is a set of *accepting states*. The DFA represents the set of sequences of labels that label any path from the initial state and ends in an accepting state.

We illustrate this definition with the example of Figure 2 by showing how to represent the constraints of a forbidden observable language. Consider $\mathcal{L}_{\text{forb}} = \{ab, aba, cb\}$, i.e., the constraint that indicates that the plan should not generate the observation ab, aba, or ca. The DFA in Figure 2 is an automaton whose language is precisely $\mathcal{L}_{\text{forb}} = \{ab, aba, cb\}$. For instance, from the initial state 0, following the sequence of observable actions aba leads to state 5 ($0 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{a} 5$) which is accepting; conversely, the sequence abd leads to non-accepting state 6 ($0 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{d} 6$). The other types of constraints ($obs(\pi) = \sigma$, π is permissible, π is impermissible) can easily be represented by a DFA too.

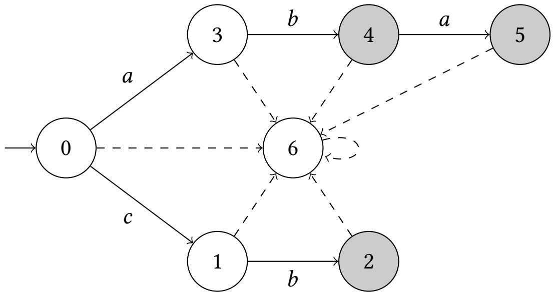

Figure 2: DFA representing the forbidden observable language $\mathcal{L}_{\text{forb}} = \{ab, aba, cb\}$. Dashed lines represent default transitions: for instance, the figure does not show any transition labelled with b from state 0; therefore the transition from 0 labelled with b leads to state 6. Shaded states are the accepting states.

A.3 Compiling DFAs into STRIPS

Given a propositional STRIPS representation of the planning problem $\mathbb{P}$ and several DFAs $\mathcal{A}_1, \ldots, \mathcal{A}_m$, we first create, for each state q of any of the DFAs, a new fact $is\_in(q)$ and add it to the set of facts $\mathcal{F}$. We then replace each action $a = \langle \text{pre}, \text{eff}^+, \text{eff}^- \rangle$ in the planning problem with the actions that we define now. Let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be all the DFA that include a in their label sets (for simplicity, we assume that their indices range from 1 to n). Then, for each combination of transitions $q_1 \xrightarrow{a} q'_1 \in T_1, \ldots, q_n \xrightarrow{a} q'_n \in T_n$, we create a new action $a_\# = \langle \text{pre}_\#, \text{eff}^+_\#, \text{eff}^-_\# \rangle$ such that:

- $\text{pre}_\# = \text{pre} \cup \{is\_in(q_1), \ldots, is\_in(q_n)\}$,
- $\text{eff}^+_\# = \text{eff}^+ \cup \{is\_in(q'_1), \ldots, is\_in(q'_n)\}$, and
- $\text{eff}^-_\# = \text{eff}^- \cup \{is\_in(q_1), \ldots, is\_in(q_n)\}$.

Essentially, this reformulation forces the planner to simulate all the DFAs as it executes the plans, so that the constraints are guaranteed to be satisfied.

Given a plan for the new planning domain, it is easy to translate this plan back for the original domain.

B PROOF OF COMPLEXITY

We prove this result by showing that conformant planning, which is ExpSpace-hard [14], can be reduced to the acceptability decision problem.

In a conformant planning instance, an action has a precondition and a set of non-deterministic effects: $\mathrm{Eff} = \{\mathrm{eff}_1, \ldots, \mathrm{eff}_k\}$, The semantics of Eff is that when the action a is applied, exactly one of the effect sets eff_j occurs, for some $j \in \{1, \ldots, k\}$. An *execution* of a sequence of actions $a_1, \ldots, a_k$ is a sequence of states $s_0, \ldots, s_k$ such that s_0 is the initial state, and each state s_i is one of the possible states reached by applying the effects of a_i in state s_{i-1}. An execution is *valid* if it ends with a specific goal action and all actions are applicable (i.e., the state s_{i-1} satisfies the precondition of a_i). The sequence of actions is a *conformant plan* if all its executions are valid.

The reduction is defined so that each execution in the conformant planning instance corresponds to one possible sequence of actions in the acceptability instance. The execution is then valid iff the sequence of actions is permissible. The non-deterministic effects are not observed in the conformant planning instance; therefore, they are implemented by non-observable actions in the acceptability instance.

Each action in the conformant planning instance is replaced by several sequences of actions in the acceptability instance. As a consequence, the state space of the acceptability instance is the Cartesian product $S \times C$ where S is the state space of the conformant planning instance and C is a set of control states that are used to enforce those sequences of actions. Let a be an action in the conformant planning instance with effects $\mathrm{Eff} = \{\mathrm{eff}_1, \ldots, \mathrm{eff}_k\}$, The acceptability instance contains the following set of actions:

$$\{a', a^\top, a^\perp\} \cup \{a^j \mid j \in \{1, \ldots, k\}\}.$$

The conditions and effects of these actions are defined in such a way that a' is always followed by one of the two actions $a^\top$ and $a^\perp$, then by one of the k actions $a^1, \ldots, a^k$. The action a' is the only observable one; its precondition is a tautology in S and it has no effect in S. $a^\top$ and $a^\perp$ are used to verify that the execution is valid: $a^\top$ is permissible and its precondition in S is the precondition of a; $a^\top$ is impermissible and its precondition in S is the negation of the precondition of a. This way, the path will be permissible iff the execution is valid. Each action a^j applies the effects eff_j. Hence, the state reached after applying one of the actions $a^1, a^2, \ldots$ will be the result of one of the non-deterministic execution of a.

So, given a sequence of actions $a_1, \ldots, a_n$ in the conformant planning instance, all possible executions of this sequence (including those that are not valid) appear in the reduced instance, and they produce the same observation $a'_1, \ldots, a'_n$. In addition, these executions are permissible iff they are valid in the original conformant planning instance.

If $a_1, \ldots, a_n$ is a conformant plan, then all plans in the acceptability instance that generate $a'_1, \ldots, a'_n$ are permissible, and they are therefore acceptable (since unambiguously permissible). If, however, $a_1, \ldots, a_n$ is not a conformant plan, then there exists an impermissible plan that generates $a'_1, \ldots, a'_n$, and that plan has the same cost as all permissible plans with the same observation; if δ is positive, those plans are therefore unacceptable since they can be mistaken for the impermissible plan. We conclude that the conformant instance admits a solution iff its reduction admits an acceptable plan with $\delta > 0$.

ACKNOWLEDGMENTS

This research is part of the *Humanising Machine Intelligence* project at ANU. We thank the project's members and the anonymous reviewers for useful discussions or comments.

REFERENCES

[1] Michael Anderson and Susan Leigh Anderson. 2007. Machine ethics: Creating an ethical intelligent agent. *Ai Magazine* 28, 4 (2007), 15–15.

[2] Fahiem Bacchus and Froduald Kabanza. 2000. Using temporal logics to express search control knowledge for planning. *Artif. Intell.* 116, 1-2 (2000), 123–191.

[3] Claire Benn. 2017. Supererogation, optionality and cost. *Philosophical Studies* 175, 10 (2017), 2399–2417.

[4] Claire Benn. 2021. Ethics must be seen to be done: Communicating compliance under uncertainty. *Manuscript submitted for publication* (2021). Available from the author upon request.

[5] Claire Benn and Alban Grastien. 2021. Reducing uncertainty in partially observed human-robot interactions. *Journal of Advanced Robotics* special issue on the Ethics, Law and Psychology of Responsibility Robotics (2021), 1–16.

[6] Tom Bylander. 1994. The Computational Complexity of Propositional STRIPS Planning. *Artif. Intell.* 69, 1-2 (1994), 165–204.

[7] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David Smith, and Subbarao Kambhampati. 2019. Explicability? Legibility? Predictability? Transparency? Privacy? Security? The emerging landscape of interpretable agent behavior. In *Proc. 29th International Conference on Automated Planning and Scheduling (ICAPS-19).* 86–95.

[8] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. 2017. Plan Explanations as Model Reconciliation: Moving Beyond Explanation as Soliloquy. In *Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI-17).* ijcai.org, 156–163.

[9] Giuseppe De Giacomo and Moshe Y. Vardi. 2013. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *Proc. 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI-13).* 854–860.

[10] Anca Dragan and Siddhartha Srinivasa. 2014. Integrating human observer inferences into robot motion planning. *Autonomous Robots* 37, 4 (2014), 351–368.

[11] Luciano Floridi and Jeff W. Sanders. 2004. On the morality of artificial agents. *Minds and machines* 14, 3 (2004), 349–379.

[12] Alban Grastien, Patrik Haslum, and Sylvie Thiébaux. 2012. Conflict-Based Diagnosis of Discrete Event Systems: Theory and Practice. In *Proc. 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR-12).*

[13] Gilbert Harman. 1986. *Change in view: Principles of reasoning.* Cambridge, Mass.: MIT Press.

[14] Patrik Haslum and Peter Jonsson. 1999. Some results on the complexity of planning with incomplete information. In *Proc. 5th European Conference on Planning (ECP-99).* 308–318.

[15] Malte Helmert. 2006. The Fast Downward Planning System. *J. Artif. Intell. Res.* 26 (2006), 191–246.

[16] Thierry. Jéron, Hervé Marchand, Sophie Pinchinat, and Marie-Odile Cordier. 2006. Supervision patterns in discrete-event systems diagnosis. In *Proc. 17th International Workshop on Principles of Diagnosis (DX-06).* 117–124.

[17] Sarah Keren, Avigdor Gal, and Erez Karpas. 2014. Goal recognition design. In *Proc. 24th International Conference on Automated Planning and Scheduling (ICAPS-14).* 154–162.

[18] Emil Keyder and Hector Geffner. 2009. Soft Goals Can Be Compiled Away. *J. Artif. Intell. Res.* 36 (2009), 547–556.

[19] Anagha Kulkarni, Siddharth Srivastava, and Subbarao Kambhampati. 2019. A Unified Framework for Planning in Adversarial and Cooperative Environments. In *Proc. 33rd AAAI Conference on Artificial Intelligence (AAAI-19).* 2479–2487.

[20] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. 2017. *A survey of methods for safe human-robot interaction.* Now Publishers.

[21] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[22] Felix Lindner, Robert Mattmüller, and Bernhard Nebel. 2020. Evaluation of the moral permissibility of action plans. *Artif. Intell.* 287 (2020), 103350.

[23] Aleck M. MacNally, Nir Lipovetzky, Miquel Ramírez, and Adrian R. Pearce. 2018. Action Selection for Transparent Planning. In *Proc. 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS-18).* 1327–1335.

[24] James H. Moor. 2006. The nature, importance, and difficulty of machine ethics. *IEEE intelligent systems* 21, 4 (2006), 18–21.

[25] Stefanos Nikolaidis, Anca Dragan, and Siddhartha Srinivasa. 2016. Viewpoint-Based Legibility Optimization. In *Proc. 11th ACM/IEEE International Conference*

on Human Robot Interation (HRI-16).

[26] Douglas Portmore. 2019. *Opting for the Best: Oughts and Options*. Oxford University Press.

[27] Miquel Ramírez and Hector Geffner. 2009. Plan recognition as planning. In *Proc. 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*. 1778–1783.

[28] Raymond Reiter. 1987. A theory of diagnosis from first principles. *Artificial Intelligence (AIJ)* 32, 1 (1987), 57–95.

[29] Jussi Rintanen. 2000. Incorporation of Temporal Logic Control into Plan Operators. In *Proc. 14th European Conference on Artificial Intelligence (ECAI-00)*. 526–530.

[30] Lindsay Sanneman and Julie A Shah. 2020. Trust Considerations for Explainable Robots: A Human Factors Perspective. *arXiv preprint arXiv:2005.05940* (2020).

[31] Silvan Sievers, Martin Wehrle, Malte Helmert, and Michael Katz. 2015. An Empirical Case Study on Symmetry Handling in Cost-Optimal Planning as Heuristic Search. In *Proc. 38th Annual German Conference on AI (KI-15) (Lecture Notes in Computer Science, Vol. 9324)*. 166–180.

[32] David E. Smith and Daniel S. Weld. 1998. Conformant Graphplan. In *Proc 15th AAAI Conference on Artificial Intelligence (AAAI-98)*. 889–896.

[33] Justin Snedegar. 2017. *Contrastive Reasons*. Oxford University Press.

[34] Wendell Wallach and Peter Asaro. 2017. *Machine ethics and robot ethics*. New York: Routledge.

[35] Benedict Wright, Robert Mattmüller, and Bernhard Nebel. 2018. Compiling Away Soft Trajectory Constraints in Planning. In *Proc. 16th International Conference on Principles of Knowledge Representation and Reasoning (KR-18)*. 474–483.