**Noname manuscript No.**
(will be inserted by the editor)

# Challenges in the Trustworthy Pursuit of Maintenance Commitments Under Uncertainty

**Qi Zhang · Edmund Durfee · Satinder Singh**

**Abstract** Cooperating agents can make commitments for better coordination, and commitments can only be probabilistic when agents' actions have uncertain outcomes in general. Our perspective is that a commitment should be made not to outcomes but to courses of action. An agent thus earns trust by acting in good faith with respect to its committed courses of action. With this perspective, we examine an atypical form of probabilistic commitments called maintenance commitments, where an agent commits to actions that avoid an outcome that is undesirable to another agent. Compared with the existing probabilistic commitment framework for enablement commitments, our maintenance commitment poses new semantic and algorithmic challenges. We here formulate maintenance commitments in a decision-theoretic setting, examine possible semantics for how agents should treat such commitments, and describe corresponding planning methods. We conclude by arguing why we believe our efforts demonstrate that maintenance commitments are fundamentally different from enablement commitments, and what that means for their trustworthy pursuit.

**Keywords** Commitment · Trust · Uncertainty

## 1 Motivation

In multiagent systems, agents are often interdependent in the sense that what one agent does can help or hinder another. In a cooperative system, agents can mutually benefit from helping each other. However, some agents could try to receive benefit without reciprocation, because helping others incurs some individual costs. If trust means having confidence that another will act so as to reciprocate help, then being trustworthy—worthy of such trust—constrains the agent to acting thusly. To persuade an agent designer to create trustworthy agents, other agents (individually and/or collectively) can form and share opinions about agents' trustworthiness, and won't act to benefit agents with a bad reputation.

Our work assumes that the designer has been persuaded. Even so, however, it isn't always clear how to create a trustworthy agent, given that an agent often

Qi Zhang · Edmund Durfee · Satinder Singh
E-mail: {qizhg, durfee, baveja}@umich.edu

lacks complete control over its environment, and is unable to predict precisely the future situations it will face. Specifically, the form of interdependency we focus on is with respect to a scenario where an agent (the commitment *provider*) makes a social commitment (Kalia et al, 2014; Singh, 1999) to another (the commitment *recipient*). When stochasticity is inherent in the environment, the provider cannot guarantee to bring about the outcomes that the recipient expects, and in fact could discover after making the commitment that how it planned to try to bring about the outcomes would be more costly or risky than it had previously realized.

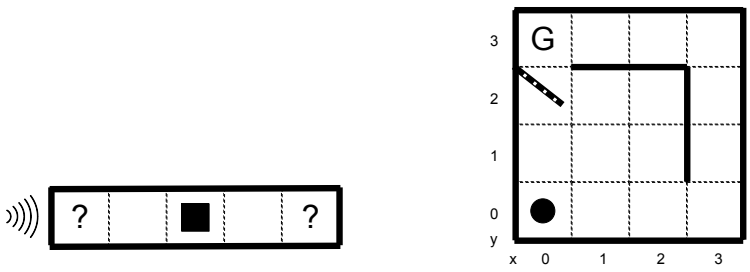
Our focus, therefore, is to define semantics and mechanisms for an agent to follow such that it is assured of faithfully pursuing its commitments despite the uncertainty. Our prior work (Durfee and Singh, 2016) articulated our perspective that such a *probabilistic* commitment should be considered fulfilled if the provider's actions would have brought about the desired outcome with a high enough expectation, even if in a particular instance the desired outcome was not realized. That is, the provider acted in good faith. Thus, even if the provider changes its course of action as it learns more about costs and risks on the fly, it can still fulfill its commitment if whatever course of action it pursued could be expected to achieve the desired outcome with at least the promised likelihood. With this perspective, previous work has focused largely on commitments of achievement (Witwicki and Durfee, 2009; Zhang et al, 2016, 2017), which we also call *enablement* commitments, where the provider commits to changing some features of the state in a way desired by the recipient with some probability by some time. For example, the recipient plans to take an action (e.g., move from one room to another) with a precondition (e.g., the door is open) that it is counting on the provider to enable.

This paper focuses on another form of commitment, which we refer to as a *maintenance commitment*, where instead of committing to some course of action that in expectation will enable conditions the recipient wants, the provider instead commits to courses of action to probabilistically avoid changing conditions that are already the way the recipient wants them maintained, up until a particular time. After that time, the condition cannot be assumed to remain unchanged, and before that time, there is a (usually small) probability it could be changed at any point. For example, an open door the recipient needs might be initially achieved, but as the provider opens and closes other doors during its housekeeping tasks, a resulting draft could close the door the recipient needs open. The provider could plan its tasks to postpone altering the riskiest doors as long as possible, but an ill-placed breeze could close the door at any time.

Our contributions in this paper are to formulate the maintenance commitment in a decision-theoretic setting, and to answer the question of how the provider and the recipient should interpret the maintenance commitment and plan accordingly. What we show is that, although superficially similar to enablement commitments, maintenance commitments impose different demands on the provider and the recipient. This in turn leads to questions about how much latitude agents can be trusted with in autonomously reacting to their uncertain environment.

## 2 Preliminaries

In this section, we describe the decision-theoretic setting we adopt for analyzing probabilistic commitments, including both enablement commitments and mainte-




(a) Provider (b) Recipient

Fig. 1: The Gate Control problem.

nance commitments. We review our prior work in the definition and semantics of enablement commitments, in preparation for our exposition on maintenance commitments in the following sections. Notation-wise, we use superscripts $p$ and $r$ to denote the provider and the recipient of a commitment, respectively.

### 2.1 The Provider's Decision-Theoretic Setting

We consider settings where the provider knows that its sequential decision making problem can be formulated in terms of Markov Decision Processes (MDPs). However, it is uncertain, at the time it is making its commitment, about the exact parameters of the MDP that truly reflects its environment. Instead, the provider knows the true MDP is one out of $K$ possible MDPs. We assume the $K$ MDPs share the same state and action spaces, but possibly have different transition and reward functions. Formally, the provider's environment is defined by the tuple $\mathcal{E}^p = \langle \mathcal{S}^p, s_0^p, \mathcal{A}^p, \{P_k^p, R_k^p\}_{k=1}^K, T^p \rangle$, where $s_0^p$ is the provider's initial state and $T^p$ is the provider's finite planning horizon. In such a finite horizon problem, states that are otherwise identical but at different time steps are different. Moreover, the provider adopts the Bayesian approach to reason about the uncertainty in its environment: it has a prior distribution $\mu_0^p$ over the $K$ MDPs, and during execution, the provider can observe the state and the reward transitions that actually occur providing information to infer the posterior distribution over the possible MDPs. When the provider learns more about potential rewards and state transitions during execution, it may be incentivized to shift to another policy that improves its utility according to the updated knowledge. The core of the problem faced by the provider is, after making a commitment to the recipient, how it should react to the evolving knowledge about the environment it is in, while still respecting the commitment.

**Simple Illustration.** In Figure 1a, we show a very simple (in this case one-dimensional) gridworld example to illustrate the formulation of the provider's problem. Here, the agent (represented by a solid black square) can move left or right, and when it reaches either end of its environment it receives a positive or negative reward. Initially, it does not know the values of the rewards at the two extremes, but it receives noisy signals about the rewards, where the signals are less noisy for a reward the closer it is to that cell. And when it is in a reward cell,

it receives the associated reward. Thus, only as the agent moves and senses does it build certainty as to which end (if either) it wants to remain in.

The environment also has a proximity sensor, at the left end, that affects a gate in the recipient's environment. The closer the provider is to the sensor, the more likely it is to trigger the sensor, which will then permanently toggle the state of the gate. If the gate is initially closed, then fulfilling a probabilistic commitment to open it will cause the provider to move leftward, but how far left it wants to move and how long it wants to linger will depend on what it learns about the rewards: if the left reward is (positive) highest, it will happily hover near the sensor and have high likelihood of opening the gate promptly, but if the right reward is (positive) highest the agent will want to limit the time and distance that it goes left. If the gate is initially open, then the provider could move to the right to limit the chances that it will prematurely close, but again how far right and for how long depends on what it learns about the desirability of the leftmost reward.

### 2.2 The Recipient's Decision-Theoretic Setting

The recipient's environment is modeled as an MDP defined by the tuple $\mathcal{E}^r = \langle \mathcal{S}^r, s_0^r, \mathcal{A}^r, P^r, R^r, T^r \rangle$, where $s_0^r$ is the recipient's initial state and $T^r$ is its finite planning horizon. While the recipient can also have uncertainty over its true MDP, for the purposes of this paper that is not necessary, and so we don't consider it.

**Simple Illustration.** Figure 1b shows the simple 2-dimensional gridworld of the recipient, where heavy solid lines indicate walls through which the agent can't move. The recipient is indicated as a solid-black dot, and it receives reward for occupying the cell marked G. Below G is the gate that is influenced by the provider. If the provider can commit, with high enough probability, to the gate being open at useful times, then the recipient can take a very direct route to G and accumulate more reward by the time horizon. Otherwise, the recipient should take the less direct (but guaranteed to be passable) route around the wall.

### 2.3 TD-POMDP framework

As one way to model the asymmetric interaction between the provider and the recipient (where the recipient depends on the provider), we use the TD-POMDP framework (Witwicki and Durfee, 2010; Zhang et al, 2016). We assume that the recipient's state can be factored as $s^r = \langle l^r, u \rangle$, where $l^r$ is the set of all the recipient's state features locally controlled by the recipient and $u$ is the set of all the recipient's state features directly affected by the provider, $u = s^p \cap s^r$. Therefore, the dynamics of recipient's state from time steps $t$ to $t+1$, given the actions of the provider and recipient ($a^p$, $a^r$ respectively) can be factored as

$$\Pr(s_{t+1}^r | s_t^r, a_t^r, s_t^p, a_t^p) = \Pr(l_{t+1}^r | s_t^r, a_t^r) \Pr(u_{t+1} | s_t^p, a_t^p). \quad (1)$$

In the original TD-POMDP framework (Witwicki and Durfee, 2010) that models the interactions between an arbitrary number of agents, each agent's local state and transition function can be factored similarly to the recipient as in (1). In this

work (and in our previous work in enablement commitments (Zhang et al, 2016)), we assume that the provider can fully control its local state features:

$$\Pr(s^p_{t+1}|s^r_t, a^r_t, s^p_t, a^p_t) = \Pr(s^p_{t+1}|s^p_t, a^p_t). \quad (2)$$

A maintenance or enablement commitment is concerned with the state features shared by both agents but only controllable by the provider, i.e. $u$. For ease of exposition, we assume $u$ is a single binary feature that can be either $u^+$ or $u^-$.

### 2.4 Enablement Commitments

Recall that our commitment semantics emphasizes the actions an agent takes (because it has control over them) instead of the states it reaches (because it cannot deterministically control them). With this semantics, then in an enablement commitment, the provider commits to pursuing a course of action that can bring about state features desirable to the recipient with some probability. Without loss of generality, we let $u^+$, as opposed to $u^-$, be the desirable state feature.

Given our semantics, one option for representing a commitment is as a policy, a mapping from the provider's state space to its action space, that the provider will follow. Based on the MDP's parameters, the commitment's probability is the probability that the policy will change $u^-$ to $u^+$. However, this representation is restricting to the provider and more opaque to the recipient. Using our example from Figure 1 to illustrate, the provider could certainly use its prior probability distribution over the possible MDPs (rewards at the 2 ends) to compute an optimal policy, but then committing to that policy means that, as it learns more about the true rewards, it would be unable to adjust its policy in response—even if doing so could be beneficial to both agents (such as if it discovers the leftmost reward is the best). And to use the commitment, the recipient would need to infer the influences the provider's policy will have on the feature $u$ that it cares about.

This suggests another option for representing a commitment. Rather than committing to a specific policy, the provider can commit to the policy's more abstract profile of probabilistic influences over time from (1). In Figure 1, for example, the influence abstraction would specify the cumulative probability of the gate being open at each time given the provider's planned movements. As the provider learns more about its environment, it can improve its local policy (to acquire more local reward) as long as it behaves within the probabilistic influence profile (where it can exceed probabilities if it wants to). In our running example, that means if it discovers the left reward appears more likely to be the best, the provider can change its policy to favor moving that way without violating its commitment.

With the influence abstraction, the recipient's problem is also simplified since it can directly encode the probability profile into its transition function, rather than having to infer it from a policy as before. Note, though, that if the provider does modify its policy, the recipient's profile might no longer accurately model the dynamics of feature $u$, as at the end of the previous paragraph where $u^+$ would be more likely earlier on (and overall). If they can intermittently communicate, the provider could update the recipient with the new profile, but then the provider would be limiting its future policy adjustments to adhering to that even more demanding commitment (e.g., if as it moves left it discovers that its beliefs about the left reward were wrong, it can't renege). In our past and current work, we

instead assume no communication at runtime, and accept the inefficiencies that arise if the recipient's model is more pessimistic relative to the provider's evolving policy. The crucial point is that the provider's latitude is only in one direction: it can't adopt a policy that underperforms relative to the commitment that it made.

Our previous work went beyond the influence abstraction to instead utilize what we called a commitment abstraction. Formally, an enablement commitment $c$ is defined by tuple $c = \langle u, T_c, p_c \rangle$, where $u$ is the condition being committed to, $T_c$ is the commitment time horizon, and $p_c$ is the commitment probability. The provider's commitment semantics is to follow a policy $\pi^p$ that, starting from its initial state $s_0^p$, has set $u$ to $u^+$ at time step $T_c$ with at least probability $p_c$, with respect to the provider's prior distribution over the $K$ MDPs. That is

$$\Pr\left(u_{T_c} = u^+ \,\middle|\, s_0^p, \pi^p, k \sim \mu_0^p\right) \geq p_c. \tag{3}$$

Here, $\pi^p$ is defined as a mapping from the provider's possible histories (including observations that revise its beliefs about which is the true MDP) to distributions over actions it should take for each. Our prior work examined the (high) computational costs of directly implementing this semantics, and developed less expensive iterative techniques that selectively expand only the parts of the history space that the provider is actually experiencing (Zhang et al, 2016, 2017).

The commitment abstraction compresses the provider's time-dependent influence on $u$ into a single timepoint, rather than a potentially more gradual profile. Why would we choose to do that? The most important reason is that it opens up even more latitude for the provider to evolve its policy as it learns more about its environment: instead of needing to meet probabilistic expectations at multiple timepoints, it can modify its policy much more flexibly as long as in the long run (by time $T_c$) it hits the target probability. Our previous work has shown the value of having such flexibility (Zhang et al, 2017). The abstraction also reduces the complexity of the recipient's reasoning, since it only needs to model a single branch (at $T_c$) for when $u^-$ probabilistically toggles to $u^+$, and assume no toggling before or afterward.[1] The cost of these gains is that the inefficiencies noted for the influence abstraction are accentuated, even in the case where the provider's policy doesn't need to evolve at all. In our running example, for instance, the gate could open before or after $T_c$, but the recipient would not model this: it will build a policy that only checks the gate at $T_c$ and goes through it if it is open, and won't consider checking or going through earlier or later. Our experiences however suggest that, if $T_c$ is chosen well, the inefficiency costs are easily outweighed by the flexibility and computational benefits.

## 3 Semantics of Maintenance Commitments

As a reminder, our maintenance commitment is motivated by scenarios where the initial value of state feature $u$ is desirable to the recipient, who wants it to maintain its initial value for some interval of time (e.g., Duff et al (2014), Hindriks and van Riemsdijk (2007)), but where the provider could want to take actions that would change it. Revisiting our running example, this is the case where the gate

[1] No toggling afterward assumes common knowledge (beyond the commitment) that once the provider achieves the condition it will never be undone before the recipient uses it.

is initially open, and the provider could cause it to be closed as a side effect of moving towards rewarding locations and triggering the sensor to toggle it.

Given our successful use of the commitment abstraction for enablement commitments, we expected to apply the same idea (and algorithms) to maintenance commitments. As we examine in the rest of this paper, fundamental differences between the two kinds of commitments make this far from straightforward.

We begin by formally defining a maintenance commitment $c$ also as a tuple $c = \langle u, T_c, p_c \rangle$, where $u$ is the commitment feature with the initial value of $u^+$. We next concentrate our exposition on the question of the semantics of commitment $c$ for the provider and the recipient, respectively.

### 3.1 Provider's Semantics

Given maintenance commitment $c = \langle u, T_c, p_c \rangle$, the provider is constrained to follow a policy that keeps $u$ unchanged for the first $T_c$ time steps with at least probability $p_c$, with respect to its prior distribution over the $K$ MDPs. Formally:

$$\Pr\left(\bigwedge_{t=1}^{T_c} u_t = u_0 \,\middle|\, s_0^p, \pi^p, k \sim \mu_0^p\right) \geq p_c, \tag{4}$$

where $\pi^p$ is the provider's policy. This is like (3), except it applies over an initial interval of time rather than at a specific timepoint. Like the semantics for enablement commitments, $\pi^p$ is a mapping from the provider's history to distributions over actions. We say that the provider's policy $\pi^p$ respects the semantics of commitment $c$ if (4) is satisfied, and later consider whether our less expensive iterative techniques for enablement commitments can apply to this case as well.

### 3.2 Recipient's Semantics

As with enablement commitments, the recipient should use the commitment tuple, $c = \langle u, T_c, p_c \rangle$, to create an approximate profile of the provider's influence on $u$, in this case regarding the probability of $u^+$ toggling to $u^-$ at various times.[2] If we were to borrow directly from the enablement formulation, we would approximate this profile as a step function, where up until time $T_c$ $u$ remains unchanged, and then beyond $T_c$ it is changed with a probability based on $p_c$. Obviously, however, this would be an overly optimistic approximation, as the recipient would be assuming perfect (rather than probabilistic) maintenance throughout the interval up to $T_c$, and thus formulate its policy based on a stronger model of the commitment than the provider's semantics (4) warrant (unless $p_c = 1$).

What we want instead is to find an approximate profile that accomplishes what we got for the enablement commitment, which never overestimates (is never overoptimistic with respect to) the true profile, with resulting potential inefficiencies arising from being too pessimistic. As we shall next see, such a profile is elusive in the case of maintenance commitments.

[2] And, as with enablement, we also use common knowledge that once maintenance fails the desired condition cannot be assumed to ever be restored.

*3.2.1 The earliest-disablement approximation*

Our first idea was to give the recipient's an approximation of the maintenance commitment's profile based on what we call the *earliest-disablement*, which can be viewed as the more correctly formulated counterpart to the latest-enablement approximation of the enablement commitment. In this approximation, the recipient assumes that, with probability $1 - p_c$, the value of $u$ will be toggled by the provider from $u^+$ to $u^-$ during the initial time step and will stay as $u^-$ thereafter; otherwise $u$ will be maintained as $u^+$ before $T_c$. Furthermore, after time step $T_c$, the value of $u$ will be permanently changed to $u^-$ with probability one. In detail:

$$\begin{aligned}
&\Pr(u_{t+1} = u^+ | u_t = u^+, c) = p_c, \quad t = 0 \\
&\Pr(u_{t+1} = u^- | u_t = u^+, c) = 1 - p_c, \quad t = 0 \\
&\Pr(u_{t+1} = u_t \ | u_t, c) = 1, \quad 0 < t < T_c \\
&\Pr(u_{t+1} = u^- | u_t = u^+, c) = 1, \quad t \geq T_c \\
&\Pr(u_{t+1} = u^- | u_t = u^-, c) = 1, \quad t \geq T_c
\end{aligned}$$

Here, the expected duration of the event $\{u = u^-\}$ is maximized, under constraint (4) prescribed by the provider's commitment semantics. This earliest-disablement approximation is simple to model, and by modeling the value of $u$ possibly changing at only 2 timepoints (times 0 and $T_c$), the recipient's planning costs are kept low.

However, it has a severe downside, which is that the approximation is not robust. During the execution of the recipient's plan derived from the earliest-disablement approximation, the recipient could end up in states that are unreachable according to its model. In our running example, for instance, the recipient's model would indicate that, if the gate ($u$) isn't closed (set to $u^-$) right after the first timestep, then it can be expected to remain open ($u^+$) all the way up to $T_c$. However, the provider may be executing a (fully commitment-satisfying given (4)) policy where the gate might close at other times between 1 and $T_c$.

A recipient using the approximation could thus find itself in a state that its model predicted to be unreachable. For example, based on the earliest-disablement approximation, the commitment recipient in the gate-control problem would check the gate status at time 1: if it is now closed then it would take the long route, but if it is still open then it would head straight for the goal through the open gate. Since its model indicates the gate cannot close before $T_c$, it can charge ahead without checking the gate. But since the gate could close earlier than $T_c$, it could unexpectedly crash into a closed gate. Or, if just in case it were to monitor the gate status, it could avoid a crash, but nevertheless find itself in uncharted territory: it will have reached a state that its model considers impossible, meaning that it cannot trust its model to create a policy that is robust in its true environment.

This was our first indication that maintenance commitments are qualitatively different, because the commitment abstraction could lead to a complete breakdown in policy execution, and not just to inefficiency like with enablements. We should ask why such breakdowns didn't happen with enablements, since arguably states can be reached that the recipient's model doesn't predict (e.g., the gate opens before $T_c$). The difference, though, was that the recipient could safely ignore an unmodeled positive transition (it just wouldn't take advantage of an early enablement), whereas it can't always safely ignore an unmodeled negative transition.

A way to fix the problem of potentially reaching unmodeled states is to expand the model to include the possibility of $u$ toggling at times other than $t = 0$ by using a small probability $\epsilon$ in place of the zero probability transitions for $t > 0$:

$$\begin{aligned}
&\Pr(u_{t+1} = u^+ | u_t = u^+, c) = 1 - \epsilon, \quad 0 < t < T_c \\
&\Pr(u_{t+1} = u^- | u_t = u^+, c) = \epsilon, \quad 0 < t < T_c \\
&\Pr(u_{t+1} = u^- | u_t = u^-, c) = 1 - \epsilon, \quad 0 < t < T_c \\
&\Pr(u_{t+1} = u^+ | u_t = u^-, c) = \epsilon, \quad 0 < t < T_c \\
&\Pr(u_{t+1} = u^- | u_t = u^+, c) = 1 - \epsilon, \quad t \geq T_c \\
&\Pr(u_{t+1} = u^+ | u_t = u^+, c) = \epsilon, \quad t \geq T_c \\
&\Pr(u_{t+1} = u^- | u_t = u^-, c) = 1 - \epsilon, \quad t \geq T_c \\
&\Pr(u_{t+1} = u^+ | u_t = u^-, c) = \epsilon, \quad t \geq T_c
\end{aligned}$$

With the $\epsilon$ probability transitions, the recipient now plans for those states previously considered unreachable according to the earliest-disablement approximation. This makes the plan robust, in that the recipient never finds itself in an unpredicted state. However, this approach eliminates the computational benefits of the commitment abstraction, because now the recipient must model branches for $u$ toggling at every possible timestep. And the costs come without any efficiency benefit, since the profile (where the toggling is most likely at the initial timestep and then only epsilon-likely afterward) can mislead the recipient into treating initial good news (e.g., the gate didn't close) as a near guarantee of success, causing it to have to backtrack more often than it might have had it not been so optimistic.

#### *3.2.2 The minimax-regret approximation*

The ending of the previous paragraph is somewhat surprising: we had posed the earliest-disablement approximation as being pessimistic, since the maintenance was modeled as lasting for the shortest possible time. Upon reflection, and as illustrated by the earlier example, the worst time for the maintenance to fail actually is *not* at the very beginning, but rather right as the condition $u^+$ (e.g., the open gate) is used. At that point, the recipient has made a maximal investment in its policy to use the commitment—an investment that goes to waste. Therefore, the recipient is incentivized to consider timepoints other than the beginning when the condition $u^+$ could change, and in particular should pay attention to those timepoints when the condition could be used.

Hence, we have considered another approximation of the maintenance commitment for the recipient, where instead the recipient tries to minimize the maximum regret of its policy. For a given recipient's policy, we can identify a worst-case scenario by modeling the provider as being adversarial, and thus leading to the maximum regret. Formally, let $\mathcal{P}_u = \{P_u(u_t, u_{t+1}) = \Pr(u_{t+1}|u_t) : P_u \text{ satisfies (4)}\}$ be the set of all possible transition functions (abstract influence profiles) of $u$ that satisfy the provider's commitment semantics (4). For any policy of the recipient $\pi^r$, an adversarial provider would pick a dynamics (profile) in $\mathcal{P}_u$ that changes $u$ at the worst possible time(s), and thus maximizing the regret of $\pi^r$:

$$R(\pi^r) = \max_{P_u \in \mathcal{P}_u} R(\pi^r, P_u) = \max_{P_u \in \mathcal{P}_u} V^*_{P_u} - V^{\pi^r}_{P_u}, \tag{5}$$

where $V^*_{P_u}$ is the optimal value function with the dynamics of $u$ being $P_u$, and $V^{\pi^r}_{P_u}$ is the value of policy $\pi^r$ when evaluated in $P_u$. With this minimax-regret formulation, the recipient's planning objective is to find $\pi^r$ that minimizes $R(\pi^r)$.

While conceptually reasonable, considering all of the provider's uncountable possible profiles in $\mathcal{P}_u$ is computationally non-trivial for the recipient. We can consider instead a finite subset of $\mathcal{P}_u$ to reduce the computation cost. Once the subset is specified, the recipient could compute the maximum regret of its policy by replacing $\mathcal{P}_u$ with that subset in (5). But which profiles in $\mathcal{P}_u$ should we consider? Note that with the earliest-disablement approximation, the recipient only considers a single profile in $\mathcal{P}_u$ that assumes failure is most likely at the very beginning and epsilon-likely afterward. The recipient could instead enumerate all such profiles in which the maintenance would fail most likely at a single timepoint before $T_c$ and epsilon-likely at every other timepoint. In our example from Figure 1, with this enumeration the recipient considers all timepoints before the commitment time that the gate could close, and finds a policy that could handle the worst possible timing. Alternatively, the recipient could include all profiles in which the maintenance would fail when it is about to use the maintained condition.

## 4 Planning and Execution with a Maintenance Commitment

In this section, we discuss the planning and execution methods used by the provider and recipient given a maintenance commitment $c = \langle u, T_c, p_c \rangle$.

### 4.1 Provider's Planning and Execution

In our previous work (Zhang et al, 2017), we examined several algorithms that an enablement commitment provider can use to plan when it is uncertain about the true MDP. In general, the provider's optimal policy will map a history of states, actions, and observations to a distribution over actions. Generating such a policy can involve massive amounts of computation, as the space of histories grows exponentially with the time horizon. Our prior work devised an algorithm we called Commitment Constrained Full Lookahead (CCFL) for computing such policies. We also developed approximate versions of this algorithm to trade-away some degree of optimality in order to dramatically reduce computation. Our Commitment-Constrained Lookahead (CCL) algorithm will only lookahead to a limited (user-specified) depth when considering how the provider's distribution over possible MDPs could change. And our Commitment-Constrained Iterative Lookahead (CCIL) will iteratively apply the CCL algorithm during policy execution, probing more deeply along only trajectories compatible with the history experienced so far.

With some effort, each of these approaches can be modified to fit maintenance commitments. The key challenge is modifying them to handle the conjunctive aspect of the commitment semantics (4) which was not present for enablement commitments (3). Without getting into details, this conjunction can be captured by enlarging the set of decision variables in the linear program used for solving such problems (Zhang et al, 2017), and the problem can be simplified by exploiting structure such as that toggling $u$ is permanent.

### 4.2 Recipient's Planning and Execution

The big difference in planning and execution with maintenance commitments is in how they affect the recipient. The earliest-disablement approximation, alone, can be treated similarly to how recipient planning is done for enablement commitments, but then execution becomes brittle, since the recipient might find itself in an unexpected state for which no actions were identified. Hence, the recipient's execution component would require modification to replan (possibly after modifying its MDP) at runtime, rather than simply just executing a policy.

To capture the epsilon-probability transitions to provide robustness, either they can be explicitly incorporated into the model, or the planning algorithm can be modified to inject them automatically. Either way, the costs of the recipient's planning will skyrocket to model all of these transitions.

The recipient's planning and execution with the simple minimax-regret approximation that picks a better time to model the transition from $u^+$ to $u^-$, with or without the epsilon-probability transitions, would be analogous to the earliest-disablement case, once the time to model the transition is picked. However, the (potentially considerable) costs for finding that better time, involving finding multiple optimal policies and adversarial responses, need to be factored in as well.

And moving towards an even more extensive profile of transition times is yet more challenging, and is still a work in progress. Because the set $\mathcal{P}_u$ is uncountable, even computing the maximum regret of a recipient's given policy $\pi^r$, i.e. $R(\pi^r)$, is nontrivial since we cannot enumerate all possible $P_u$. We want to reduce the size of $\mathcal{P}_u$ without too much approximation error when computing maximum regret.

## 5 Discussion

In this paper, we described our experiences in taking what we learned about semantics and algorithms for trustworthy fulfillment of enablement commitments, and attempting to map it to maintenance commitments. Hopefully, we've convinced the reader that such a mapping is nontrivial, which suggests that, despite their similarities in describing the toggling of conditions over time, maintenance commitments are fundamentally different from enablement commitments.

One answer to these differences, that we've focused on here, is to modify solutions that work best for enablement commitments, based on the (single-timepoint) commitment abstraction, to be better suited to maintenance commitments. We are currently evaluating the effectiveness of these modifications empirically.

An alternative answer, that may have ramifications more broadly to issues of trustworthy social commitments in other (non-decision-theoretic) settings, is to question whether an abstract commitment is sensible when it comes to maintenance as opposed to enablement. The introduction of epsilon-probability branches for robustness suggests that, if the recipient needs to model more branches for robustness anyway, why not have these branches more properly reflect the real profile over the maintenance interval, so the recipient formulates a policy that performs better in the world it will actually experience?

Recall our progression when we talked about enablement commitments in Section 2.4, where we went from commitments to policies, to abstract influences, to the commitment abstraction. There, the commitment abstraction gave the provider

more flexibility, and gave the recipient computational advantages at a generally minor policy efficiency cost. What we've uncovered is that, for maintenance commitments, the recipient cannot safely get the computational advantages of the commitment abstraction, which then raises the question of whether we should back up, and use a less abstract representation, like the influence abstraction, when modeling maintenance commitments. This will constrain the provider more, but our explorations in this paper suggest that this may be unavoidable: that when it comes to maintenance, useful commitments need to be more fine-grained, and will more tightly restrict the flexibility of a provider in an uncertain environment.

In conclusion, our goal for this paper is to spur reflection and discussion about trust in the context of different kinds of expectations (specifically enablement versus maintenance). How (especially in non-decision-theoretic settings) do agents express expectations about each other for these different needs? How is trust measured, or failure to be trustworthy detected, in such cases? Should agents have reputations not only for what they can (or can't) achieve for each other, but also for how reliably they can avoid interfering with each other, and if so are these reputations established/evaluated differently?

## Acknowledgements

We thank the anonymous reviewers for their helpful suggestions. Supported in part by the US Air Force Office of Scientific Research, under grant FA9550-15-1-0039, and by the Open Philanthropy Project to the Center for Human-Compatible AI.

## References

Duff S, Thangarajah J, Harland J (2014) Maintenance goals in intelligent agents. Computational Intelligence 30(1):71–114

Durfee EH, Singh S (2016) On the trustworthy fulfillment of commitments. In: Osman N, Sierra C (eds) AAMAS Workshops (Selected Papers), Springer International Publishing, pp 1–13

Hindriks KV, van Riemsdijk MB (2007) Satisfying maintenance goals. In: 5th Int. Workshop Declarative Agent Languages and Technologies (DALT), pp 86–103

Kalia AK, Zhang Z, Singh MP (2014) Estimating trust from agents' interactions via commitments. In: 21st Euro. Conf. on AI (ECAI), pp 1043–1044

Singh MP (1999) An ontology for commitments in multiagent systems. Artificial Intelligence and Law 7(1):97–113

Witwicki SJ, Durfee EH (2009) Commitment-based service coordination. IntJ Agent-Oriented Software Engineering 3:59–87

Witwicki SJ, Durfee EH (2010) Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In: Int. Conf. Auto. Planning Sys. (ICAPS), pp 185–192

Zhang Q, Durfee EH, Singh SP, Chen A, Witwicki SJ (2016) Commitment semantics for sequential decision making under reward uncertainty. In: Int J. Conf. on Artificial Intelligence (IJCAI), pp 3315–3323

Zhang Q, Singh S, Durfee E (2017) Minimizing maximum regret in commitment constrained sequential decision making. In: Int. Conf. Automated Planning Systems (ICAPS), pp 348–356