Minimizing Maximum Regret in Commitment Constrained Sequential Decision Making

Qi Zhang, Satinder Singh, Edmund Durfee
Computer Science and Engineering, University of Michigan
{qizhg,baveja,durfee}@umich.edu

Abstract
In cooperative multiagent planning, it can often be beneficial for an agent to make commitments about aspects of its behavior to others, allowing them in turn to plan their own behaviors without taking the agent’s detailed behavior into account. Extending previous work in the Bayesian setting, we consider instead a worst-case setting in which the agent has a set of possible environments (MDPs) it could be in, and develop a commitment semantics that allows for probabilistic guarantees on the agent’s behavior in any of the environments it could end up facing. Crucially, an agent receives observations (of reward and state transitions) that allow it to potentially eliminate possible environments and thus obtain higher utility by adapting its policy to the history of observations. We develop algorithms and provide theory and some preliminary empirical results showing that they ensure an agent meets its commitments with history-dependent policies while minimizing maximum regret over the possible environments.

Introduction
When planning jointly, agents can benefit from making commitments to each other about what they will (or won’t) do that affects another agent, so that other agents can form their own plans accordingly. In the ideal case, commitments by an agent could allow the other agents to plan their behaviors completely independently by relying on the commitments. For example, an agent could commit to free up a tool for another agent to use by a certain time and, assuming that the only interaction among the two agents is the use of the tool, this can allow the other agent to plan independently.

Some existing computational models of commitments characterize them using formal logic (Cohen and Levesque 1990; Castelfranchi 1995; Singh 1999; Mallya and Huhns 2003; Chesani et al. 2013; Al-Saqqar et al. 2014). When there is uncertainty about the consequences of actions, logical formulations associate conventions and protocols for managing such uncertainty (Jennings 1993; Xing and Singh 2001; Winikoff 2006). An alternative means of handling uncertainty, as in this paper, is to formalize commitments in decision-theoretic settings and explicitly allow for probabilistic guarantees of outcomes (Xuan and Lesser 1999; Bannazadeh and Leon-Garcia 2010; Witwicki and Durfee 2009).

An interesting challenge in making and keeping commitments arises when the committing agent expects to learn information about its environment while executing its plan. What should a probabilistic commitment mean in such a setting? Recently we (Zhang et al. 2016) provided an answer to this question in sequential decision problems where the committing agent interacts with an environment modeled as a controlled Markov process with a prior distribution over possible reward functions, and has already made a probabilistic commitment to achieve a state at a certain time. The committing agent observes rewards while taking actions and thereby can refine its distribution over possible reward functions after each action. We formalize the meaning of a probabilistic commitment as requiring the agent to “execute a policy from the initial state that properly affects the committed state variables in expectation” (where this expectation is over both stochastic transitions and the effect of stochastic reward observations on the agent’s knowledge during plan execution).

Our main contributions in this paper are to extend our work to the worst-case non-Bayesian setting in which the agent knows that the sequential decision making task it is facing is from one of a set of Markov Decision Processes (MDPs), where both reward and transition dynamics could differ across MDPs, and nonetheless guarantees, at least, the same commitment probability in all MDPs. We propose a family of policy construction methods for the committing agent that adopts maximum regret as the performance criterion. We prove that policies constructed by the proposed methods respect this commitment semantics, and through experimental results we find they significantly outperform some baseline policies, such as the greedy policy that picks the next action minimizing myopic regret.

Example Domain
For illustrative purposes, we first present a two-state example, Twin-States, before we formalize the general problem. The Twin-States domain consists of two states with known deterministic transition dynamics but uncertain reward, as shown in Figure 1. The start state is A and the agent has three actions in each of the two states. Action $a_0$ moves the agent to the other state with no reward, while actions $a_1$ and...
that it cannot be in. Formally, we can summarize the current history $h$ into a knowledge state, $b := \langle s, \kappa \rangle$, where $s$ is the current env-state, and $\kappa := \{ k : k \sim h \}$ is the set of indices of MDPs consistent with $h$. Initially, the agent is in knowledge state $b_0 = \langle s_0, \kappa_0 \rangle$ where $\kappa_0 = \{ 1, 2, \ldots, K \}$. Let $B_t$ be a random variable indicating the knowledge state at time step $t$, and $b_t$ be the knowledge state given history $h_t$. (In general there is a many to one mapping from histories to knowledge states.) We define the agent’s planning objective below.

**Commitment Semantics**

Note that there are two types of uncertainty in our setting. There is non-probabilistic uncertainty (i.e., incomplete knowledge) over which MDP the agent is facing, and there is probabilistic uncertainty (i.e., stochastic state transitions and possibly rewards) within an MDP.

Our commitment in this uncertain environment $E$ is formally defined as follows.

**Definition 1.** A probabilistic commitment $c$ is formally defined as a tuple $(\Phi, T, p)$, where $\Phi \subset \mathcal{S}$ is the commitment env-state space, $T$ is the commitment finite time horizon, and $p$ is the commitment probability. By making commitment $c$, the agent is constrained to follow a policy $\pi$, such that

$$
\Pr_\pi (S_T \in \Phi | S_0 = s_0; k) \geq p, \forall k \in \kappa_0.
$$

From Equation (2), the semantics of a probabilistic commitment is clear: the agent is constrained to follow a (in general history-dependent) policy, such that starting at the initial env-state, it will reach an env-state in the committed env-state space, $S_T$, at the time horizon, $T$, with at least the committed probability, $p$, no matter which MDP it is in. Given probabilistic commitment $c$, let $\Pi_c$ be the set of all history-dependent stochastic policies that satisfy Equation (2).

**Minimax Regret**

In this paper, we are interested in finding a good policy given a probabilistic commitment, using maximum regret as the performance criterion. Let

$$
U^\pi(k) = E_\pi \left[ \sum_{t=0}^{T-1} R_{S_t, A_t}(k) | S_0 = s_0; k \right]
$$

be the expected cumulative reward under policy $\pi$ if the true MDP is $k$, and let $U_c^*(k)$ be the expected cumulative reward under the optimal policy respecting the semantics of commitment $c$ if the true MDP is $k$:

$$
U_c^*(k) = \max_{\pi \in \Pi_c} U^\pi(k).
$$

Finding $U_c^*(k)$ amounts to solving a standard constrained MDP problem and this can be done efficiently by linear programming (Altman 1999). Given commitment $c$, let $\rho_c^*$ denote the maximum regret of policy $\pi$ under $c$, i.e.,

$$
\rho_c^* = \max_{k \in \kappa_0} U_c^*(k) - U^\pi(k).
$$

Let $\Pi_c^*$ be the set of policies that minimizes the maximum regret while respecting the commitment semantics,

$$
\Pi_c^* = \{ \pi : \pi \in \Pi_c, \rho_c^* = \min_{\pi' \in \Pi_c} \rho_{\pi'}^* \}.
$$
The agent’s planning goal is to find a policy in Π∗. We conclude this section with a series of formal observations showing that straightforward planning methods will not be enough to construct policies in Π∗.

Observation 1. Let π∗(k) = arg maxπ∈Πk Uπ(k) be a policy respecting commitment c that is optimal if the true MDP is k. Then, in general we have π∗(k) /∈ Π∗, ∀k.

Observation 2. Let πG be the greedy policy under which the agent selects the next action that minimizes the maximum myopic regret.

Observation 3. There exists an environment E where all MDPs are deterministic, i.e., ∃k, s, a ∃s′ such that Psa(k) = 1, and no policy in Π∗ is deterministic.

The Twin-States domain provides a proof of the above observations by example as we verify in the section of Empirical Results below.

Finally, we might think whenever the agent learns more about the true MDP during execution it is a good idea to re-plan from the current env-state with the original commitment probability. Clearly, if during execution one can always find a policy that achieves the original commitment probability conditioned on the current env-state, such a re-planning approach will certainly respect the commitment semantics. Observation 4 says that this is not always possible, and the example shown in Figure 2 verifies it.

Observation 4. There exists π ∈ Π such that if the agent executes policy π for the first t > 0 time steps starting in state s0, the history generated, hπ, is such that

∀π′, ∃k ∈ κt Pπ′(ST ∈ Φ|St = st; k) < p.

Methods

In this section we introduce several methods for constructing policies that respect the commitment semantics for a given commitment c.

Commitment Constrained No-Lookahead

Let Π0 be the set of all Markov policies, i.e., policies that choose actions solely as a function of the current env-state (and ignore κ). Assuming Π0 ∩ Π∗ = ∅, our Commitment Constrained No-Lookahead (CCNL) method of Figure 3 finds a minimax regret Markov policy respecting the

$$\min_{x, k} U_c^*(k) - U(k) \quad (3a)$$

subject to

$$\forall k \hspace{1cm} U(k) = \sum_{s, a} x_{sa}(k) R_{sa}(k) \quad (3b)$$

$$\forall k, s, a \hspace{1cm} x_{sa}(k) \geq 0 \quad (3c)$$

$$\forall k, s, a \hspace{1cm} x_{sa}(k) = \sum_{s, a} x_{sa}(k) P_{sa}(k) + \delta_{s's0} \quad (3d)$$

$$\forall k, k', s, a \hspace{1cm} \sum_{a'} x_{sa'}(k') \geq 0 \quad (3e)$$

$$\forall k, s \in \Phi \sum_{a} x_{sa}(k) \geq p \quad (3f)$$

Figure 3: CCNL program. It uses occupancy measures x as decision variables. Constraint (3b) guarantees that U(k) is the cumulative reward in MDP k, through which the maximum regret is expressed in objective function (3a). Constraints (3c) and (3d) guarantee that x(k) is a valid occupancy measure given that the initial state is s0 and the transition function of the kth MDP is P(k), where δs's0 is the Kronecker delta that returns 1 when s' = s0 and 0 otherwise. Constraint (3e) guarantees that all K occupancy measures have the same underlying Markov policy. The commitment semantics is explicitly expressed in constraint (3f). The corresponding Markov policy can be recovered via Equation (5) in the main text.
commitment semantics, which is a solution to the following problem:

$$\min_{\pi \in \Pi_c \cap \Pi_L} \rho_c^\pi. \quad (4)$$

For MDP $k$, each policy $\pi$ has a corresponding occupancy measure $x^\pi(k)$ for env-state-action pairs:

$$x^\pi_{sa}(k) := E_\pi \left[ \sum_{t=0}^{T-1} \mathbf{1}\{S_t=s, A_t=a\} | S_0 = s_0; k \right].$$

We will use shorthand notation $x(k)$ in place of $x^\pi(k)$ when policy $\pi$ is clear from the context. If $\pi$ is a Markov policy, it can be recovered from its occupancy measure via

$$\pi(a|s) = \frac{x_{sa}(k)}{\sum_{a'} x_{sa'}(k)}. \quad (5)$$

Figure 3 presents our straightforward adaptation of the linear program for finding constrained-optimal policies (Altman 1999) in MDPs (see the caption of Figure 3 for details).

**Commitment Constrained Lookahead**

During execution, the agent can observe the env-state transitions and reward, and reason about the true MDP it might be in or, equivalently, the MDPs that it cannot be in. Thus, restricting the agent to Markov policies as in the previous section will lead to larger regret than is necessary. Here we consider the general case where the agent may choose actions based on the knowledge state (or equivalently history) for the first $0 < L \leq T$ steps, and use the env-state for the remaining time steps (if $L = 0$, we recover the Markov policy case above). We refer to $L$ as the knowledge-state-update boundary. The resulting $L$-updates policy has the form:

$$\pi(a|s) = \begin{cases} \pi(a|b_t) & t < L \\ \pi(a|s_t, b_L) & t \geq L, \end{cases}$$

where $b_t$ is the knowledge state consistent with $h_t$, and $b_L$ is the knowledge state consistent with $h_L$ when $t \geq L$. It is important to note that, after the knowledge-state-update boundary, the policy conditions on both the env-state as well as the last updated knowledge state $b_L$

For example, Figure 4 shows a ($L = 1$)-updates policy constructed in the Twin-States domain. After taking some action in the initial knowledge state, depending on which knowledge state it actually ends up in at time $L = 1$, it then executes a Markov policy, represented by a curve, all the way up to the horizon. Those Markov policies starting from time step $L = 1$ are not necessarily the same, which gives the agent flexibility of choosing different behaviors based its updated knowledge about the environment.

Let $\Pi_L$ be the set of all $L$-updates policies. Our Commitment Constrained Lookahead (CCL) method finds a minimax regret $L$-updates policy respecting the commitment semantics, which is a solution to the following problem:

$$\min_{\pi \in \Pi_c \cap \Pi_L} \rho_c^\pi. \quad (6)$$

Problem (6) can be expressed by the program in Figure 5.

The program in Figure 5 introduces as decision variables $y(k)$ and $x(k)$ for every possible MDP $k$, where $y(k)$ is the knowledge state-action occupancy measure if the true MDP is $k$, but only for those knowledge states reachable within the first $L$ time steps, and $x(k)$ is the env-state-action occupancy measure for the env-states in the remaining $T - L$ time steps if the true MDP is $k$. See the caption of Figure 5 for details.

Any $L$-updates policy $\pi_L$ respecting the commitment semantics can be derived from a feasible solution to the program in Figure 5 via

$$\pi_L(a|h_t) = \begin{cases} \pi_L(a|b_t) = \frac{y_{b_t}(k)}{\sum_{a'} y_{b'a'}(k)} t < L \\ \pi_L(a|s_t, b_L) = \frac{y_{s_t}(k)}{\sum_{a'} y_{s'a'}(k)} t \geq L. \end{cases} \quad (8)$$

Theorem 1 states that CCL with knowledge-state-update boundary $L$ finds a minimax regret policy in $\Pi_c \cap \Pi_L$.

**Theorem 1.** If $\Pi_c \cap \Pi_L \neq \emptyset$ holds for commitment $c$, the program in Figure 5 is feasible. Let $x^*$, $y^*$ be its optimal solution, then the policy derived via Equation (8) with $x^*$, $y^*$ is a minimax regret policy in $\Pi_c \cap \Pi_L$.

The proofs for Theorem 1 and the theorems that follow are presented in the Appendix of a full version of this paper available on arXiv.

Intuitively, a knowledge-state-update boundary greater than zero may help the agent choose actions according to its changing knowledge about the actual MDP it is in and therefore improve the performance. Theorem 2 says the maximum regret of the policy derived by CCL using any $L > 0$ is upper bounded by the maximum regret of the policy derived by CCNL.

**Theorem 2.** If $\Pi_c \cap \Pi_0 \neq \emptyset$ holds for commitment $c$, the program in Figure 5 is feasible for any $L \in [0, T]$. Let $\pi_L^*$ be the policy derived by CCL using knowledge-state-update boundary $L$, then for any $L \in [0, T]$ we have

$$\rho_c^\pi_L \leq \rho_c^\pi_0.$$
\[
\begin{align*}
\min_{x,y} & \max_{k \in \kappa_0} \ U_*^c(k) - U(k) \\
\text{subject to} & \\
\forall k \in \kappa_0 & \\
U(k) = & \sum_{b \in B^k_{\{0,L\}}; a} y_{ba}(k) \tilde{R}_{ba}(k) + \sum_{b,l \in B^k_{S}; s,a} x^{bL}_{sa}(k) R_{sa}(k); \\
\forall k, b, a & \text{ } y_{ba}(k) \geq 0; \\
\forall k, b' \in \kappa & \text{ } \big(\langle s', \kappa' \rangle \big) \in B^k_{S} \\
\sum_{a'} y_{ba'}(k') = & \sum_{a} y_{ba}(k) \big(\sum_{a'} y_{ba'}(k') \big) \\
\forall k, b, a, L \in B^k_{\{0,L\}}, s & \text{ } x^{bL}_{sa}(k) \geq 0; \\
\forall k, b \in B^k_{\{0,L\}} & \text{ } \big(\langle s', \kappa' \rangle \big) \in B^k_s, s' \\
\sum_{a'} x^{bL}_{sa'}(k) = & \sum_{a} x^{bL}_{sa}(k) \big(\sum_{a'} x^{bL}_{sa'}(k') \big) + y_{ba}(k) \big(\sum_{a'} \delta_{s'a'} \big) \\
\forall b \in B^k_{\{0,L\}} & \text{ } \big(\langle s', \kappa' \rangle \big) \in B^k_s, s' \\
\sum_{a'} x^{bL}_{sa'}(k) = & \sum_{a} x^{bL}_{sa}(k) \big(\sum_{a'} x^{bL}_{sa'}(k') \big) \\
\forall k \in \kappa_0 & \sum_{b \in B^k_{\{0,L\}}; s \in \Phi, a} x^{bL}_{sa}(k) \geq p
\end{align*}
\]

(minimax regret objective) \hspace{1cm} (7a)

(utility if MDP \( k \) is true) \hspace{1cm} (7b)

(policies via \( y(k) \) and \( y(k') \) are consistent) \hspace{1cm} (7c)

(define \( y_{ba} \) as the prob of reaching \( b_L \)) \hspace{1cm} (7d)

(policies via \( x(k) \) and \( x(k') \) are consistent) \hspace{1cm} (7e)

(commitment semantics) \hspace{1cm} (7f)

Theorem 3. There exists an environment \( E \), a commitment \( c \), \( L' > L > 0 \) satisfying \( \Pi_c \cap \Pi_L \neq \emptyset \) and \( \Pi_c \cap \Pi_{L'} \neq \emptyset \), such that

\[
\rho_c^* > \rho_c^L,
\]

where \( \pi_c^* \) and \( \pi_c^L \) are the policies derived by CCL using boundaries \( L \) and \( L' \), respectively.
Theorem 4. If the transition dynamics does not vary across MDPs in environment $\mathcal{E}$, i.e. $\forall k, k', P(k) = P(k')$, and $\Pi_c \cap \Pi_L \neq \emptyset$ for boundary $L$, then for any $L' > L$ we have $\Pi_c \cap \Pi_{L'} \neq \emptyset$, and
\[ \rho^{\pi^*_L}_{c} \leq \rho^{\pi^*_L}_{c}, \]
where $\pi^*_L$ and $\pi^*_{L'}$ are the policies derived by CCL using boundaries $L$ and $L'$, respectively.

Commitment Constrained Iterative Lookahead

Commitment Constrained Iterative Lookahead (CCIL), as the name suggests, iteratively applies the CCL technique during execution. Suppose starting from the initial knowledge state the agent executes the first $L$ actions prescribed by a minimax regret $L$-updates CCL policy $\pi^*_{L}$ derived by solving the program in Figure 5 and ends up in knowledge state $b_L \in B_{L}^{b_0}$. Instead of executing the remaining $T-L$ actions prescribed by $\pi^*_{L}$, the agent can re-construct a new $L$-updates policy with an initial knowledge state now $b_L$. This policy reconstruction is helpful because the agent gets more knowledge about the true MDP by observing the transitions and reward in the first $L$ steps. Due to the changed initial knowledge state, naively sticking with the original commitment probability might lead to the difficulty stated in Observation 4. To respect the commitment semantics, the agent should instead plan with a commitment probability updated as follows. Let $b_L = (s_L, \kappa_L)$, where $s_L$ is the current env-state, and $\kappa_L$ is the set of MDPs consistent with the history up to time step $L$. For every possible MDP $k \in \kappa_L$, the agent will commit the probability as the achieved probability if the agent were to stick with $\pi^*_{L}$ from $s_L$:
\[ p(k) = \Pr(T \in \Phi|S_L = s_L; k) . \]

Then, the agent can construct a new $L$-updates policy by solving the program in Figure 5 with the following modifications:

1. Start from current knowledge state $b_L$ instead of $b_0$, i.e. replace every $b_0$ with $b_L$, and $\kappa_0$ with $\kappa_L$ in the program.
2. Plan with the updated commitment probabilities, i.e. replace $p$ in the last constraint of the program with $p(k)$ calculated as Equation (9).
3. Replace $U^*_c(k)$ with $U^*_{s_L, p(k)}(k)$ which is defined as the optimal objective value of the following problem:
\[ \max_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=L}^{T-1} R_{s_t, a_t} | S_L = s_L; k \right] \]
\[ \text{subject to } \Pr(T \in \Phi|S_L = s_L; k) \geq p(k) \]
which is the expected cumulative reward of the optimal policy that achieves commitment probability $p(k)$ from current env-state $s_L$ in MDP $k$.

This modified program is guaranteed to be feasible because the original $L$-updates policy $\pi^*_{L}$ itself is a solution. CCL iteratively applies the above procedure every $L$ steps. We outline CCIL in Algorithm 1, and Theorem 5 formally states that it respects our commitment semantics.

Theorem 5. If $\Pi_c \cap \Pi_L \neq \emptyset$ holds for commitment $c$ and boundary $L > 0$, let $\pi^*_{L}$ be the history-dependent policy defined as Algorithm 1. We have $\pi^*_{L} \in \Pi_c$, i.e., CCIL respects the commitment semantics.

Algorithm 1: CCIL

Input: Environment $\mathcal{E} = (S, A, s_0, \{P(k), R(k)\}_{k=1}^K)$, commitment $c = (\Phi, T, p)$, integer $L \in (0, T]$ such that $\Pi_c \cap \Pi_L \neq \emptyset$;
1. $b_0 \leftarrow (s_0, \kappa_0)$;
2. $\pi_0 \leftarrow L$-updates policy derived by solving the program in Figure 5;
3. $t \leftarrow 0$;
4. while $t < T$ do
  5.     for $i = 1, 2, ..., L$ do
  6.         Take action $a_t \sim \pi_t(\cdot|b_t)$ and observe reward-next state transition $(s_t, a_t, r_t, s_{t+1})$;
  7.         Update knowledge state as $b_{t+1} = (s_{t+1}, \kappa_{t+1})$;
  8.         $\pi_{t+1} \leftarrow \pi_t$;
  9.         $t \leftarrow t + 1$;
10. while $t < T$ do
11.     for $k \in \kappa_t$ do
12.         $p(k) \leftarrow \Pr_T(S_T \in \Phi|S_t = s_t; k)$;
13.         $U^*_{s_L, p(k)}(k) \leftarrow$ optimal objective value of (10);
14. end
15. $\pi_t \leftarrow$ policy derived by solving a modified version of the program in Figure 5; replacing every $b_0$ with $b_t$, $\kappa_0$ with $\kappa_t$, $p$ with $p(k)$, and $U^*_c(k)$ with $U^*_{s_L, p(k)}(k)$;
16. end

MILP Formulation

The CCL program in Figure 5 introduces quadratic equality constraints (7e) and (7i) to ensure that the action selection rules derived from occupancy measures in all possible MDPs are identical. These constraints make the optimization problem non-convex and hard to solve. In practice, many math-programming solvers are unable to handle programs with quadratic equality constraints. Although some solvers can deal with such programs, they often need to take as input a feasible solution as the starting point, but finding a feasible solution by itself might be difficult, and the final solutions are usually sensitive to starting points. Here we introduce a straightforward modification to the CCL program in Figure 5 that replaces the quadratic equality constraints with mixed integer constraints, and therefore reformulates it into a Mixed Integer Linear Program (MILP) that has many available solvers. The cost of this reformulation is that the derived policy is restricted to be deterministic.

Specifically, we introduce indicators $\Delta$ into the CCL program in Figure 5 as additional decision variables with the following constraints:
\( \forall b \in B_{[0, L]}^b, a \Delta_{ba} \in \{0, 1\}; \) (choose \( a \) in \( b \) iff \( \Delta_{ba} = 1 \))
\( \forall b \in B_{[0, L]}^b, a \sum_a \Delta_{ba} \leq 1; \) (at most one action is chosen)
\( \forall k, b \in B^a_{[0, L]}, a y_{ba} (k) \leq \Delta_{ba}; \) (\( y \) is consistent with \( \Delta \))
\( \forall b_k, B_{[0, L]}^b, a, a \Delta_{ba} \in \{0, 1\}; \) (choose \( a \) in \( s \) iff \( \Delta_{sa} = 1 \))
\( \forall b_k, B_{[0, L]}^b, a \sum_a \Delta_{ba} \leq 1; \) (at most one action is chosen)
\( \forall k, b_L, B_{[0, L]}^b, a, x_{ba}^L (k) \leq \Delta_{ba}^L; \) (\( x \) is consistent with \( \Delta \)).

Then, any feasible solution with the above constraints replacing constraints (7e) and (7i) of the program in Figure 5 yields a deterministic policy via Equation (8), which can be alternatively expressed using the indicator variables:

\[
\pi_L (a | h_t) = \begin{cases} 
\pi_L (a | h_t) = 1_{\{\Delta_{h_t} = 1\}} & t < L \\
\pi_L (a | s, b_L) = 1_{\{\Delta_{s}^{b_L} = 1\}} & t \geq L.
\end{cases}
\]  

(11)

Note that the objective function of the program in Figure 5 is non-linear due to the max operator. However, it is easy to reformulate it into a linear objective function with a set of linear constraints. In particular, one can introduce a scalar variable \( z \) to replace the objective function (7a) with

\[
\min_{x, y, z} \quad z
\]

and add the following constraints on \( z \)

\[ \forall k \in \kappa, z \geq U^* (k) - U (k). \]

With the above modifications, the program in Figure 5 becomes a MILP. The derived policy via (11) using an optimal solution to this MILP is a deterministic policy that minimizes the maximum regret of all deterministic policies in \( \Pi_c \cap \Pi_L \) (assuming this intersection is non-empty).

**Empirical Results**

We evaluate the performance of CCL and CCIL, under various choices of the boundary \( L \), first on the Twin-States domain of Figure 1 that has uncertain rewards, and second on the Slippery T-Maze gridworld domain of Figure 7 that has uncertain transition dynamics. CCL and CCIL MILP programs are solved using CPLEX 12.6.

**Results on the Twin-States Domain**

The main goals of the experiments on this domain are 1) to provide a constructive proof of Observations 1 to 3, 2) to evaluate the loss of the MILP formulation in a domain where an exact stochastic CCL policy can be computed, and 3) to compare the performance of CCL and CCIL using various boundaries against simple policy construction methods.

**Short horizon.** Here we set the time horizon to two so that we can find an exact stochastic minimax regret CCL policy\(^1\) and compare it with that found using the MILP formulation.

\(^1\)This exact policy is found not by solving the program in Figure 5 but as follows. Note that with only two actions available, the agent should not move to state \( B \) because it has to move back to state \( A \) using the second action and will get no reward at all. We

Figure 6 plots the maximum regret under various choices of boundary \( L \) using exact CCL, MILP-CCL, and MILP-CCIL. Because exact CCL achieves better performance than MILP-CCL, it is clear that the derived policy must be stochastic, which provides a constructive proof of Observation 3.

**Longer horizon.** Here we are concerned with comparing MILP-CCL and MILP-CCIL against the following baseline policy construction methods mentioned in Observation 1 and Observation 2 under longer than 2 time horizon.

- **MDPs-Best:** First find the optimal policies respecting the commitment semantics for every possible MDP, i.e. \( \pi_k = \arg \max_{\pi \in \Pi} U^* (k) \). The MDPs-Best policy is the one out of \( \{ \pi_k \}_{k=1}^K \) that minimizes the maximum regret.

- **Greedy:** Select the next action that minimizes the maximum one-step myopic regret over the possible MDPs consistent with the current history, i.e.

\[
a_t = \arg \min_{a_t} \max_{k \in \kappa, a_t \in A_t} \{ \max_{k \in \kappa} R_{s_t} (a_t (k)) - R_{s} a (k) \}
\]

where \( A_t \) is the set of actions available at time \( t \) that are chosen to guarantee the commitment semantics is respected. For this domain, we let \( A_t = \{ a_0, a_1, a_2 \} \) if \( t < T - 1 \). When \( t = T - 1 \), i.e. for the last action, \( A_t = \{ a_0 \} \) if \( s_t \) is \( B \), or \( A_t = \{ a_1, a_2 \} \) if \( s_t \) is \( A \).

Table 1 summarizes the results. For MILP-CCL, performance is monotonic. It takes three steps to resolve the reward uncertainty by taking action \( a_2 \) in state \( A \), moving to exploit this fact to compute an exact stochastic CCL policy, i.e. an exact solution to the program in Figure 5 by solving another equivalent mathematical program: 1) Introduce \( \pi (a | b), \pi (a | s; b) \) as decision variables, which are the probability of choosing action \( a \) under an \( L \)-updates policy for \( t < L \) and \( t \geq L \), respectively. Because choosing \( a_0 \) is sub-optimal, we need to only consider \( a \in \{ a_1, a_2 \} \). 2) Express the maximum regret as the objective function, the only constraint is that \( \pi (a | b), \pi (a | s; b) \) should be valid probability measures. The commitment semantics is automatically satisfied because we don’t need to include action \( a_0 \).

Figure 6: Maximum regret in the Twin-States domain of exact CCL, MILP-CCL, and MILP-CCIL when horizon is two. Markers “*” for MILP-CCIL overlap with makers “o” for MILP-CCL when \( L = 1, 2 \). (Note that MILP-CCIL is not defined for \( L = 0 \).)
state B, and then taking action \(a_2\) again. This explains why \(L\) larger than three does not improve the performance. If the horizon is large enough, the agent should explore the reward of action \(a_2\) in both states, then execute the action with the highest reward before going back to state A to respect the commitment semantics. We find that is exactly what MILP-CCL(\(L \in [3, T]\)) and MILP-CCIL(\(L = 1\)) do when horizon \(T \geq 7\), which causes a max regret of 5 when reward of \(a_2\) is the lowest (i.e., 1 in state A and 0 in state B).

### Results on the Slippery T-Maze

The main goals of the experiments reported here are to evaluate CCL and CCIL with the MILP formulation in a domain where 1) the transition dynamics are uncertain, and 2) the commitment probability is less than one, and thus stochastic action selection is more likely to be crucial to achieving better performance. The domain consists of two corridors that are connected as shown in Figure 7. The agent starts in the cell with a black dot and can move in four directions. Staying in cell “r” results in a positive unit of reward every time step, but the agent commits to being in cell “c” at the time horizon. There are an uncertain number of consecutive slippery cells between cell “s” and the black dot cell. In a slippery cell movement actions succeed with probability .8. Cell “s” is known to be slippery. The agent does not know in advance the number of slippery cells, which makes the transition dynamics uncertain.

Figure 8 shows the results under commitment time horizon \(T = 10\) and commitment probability \(p = 0.6\). The maximum regret of MILP-CCL is equal to the objective value of the mathematical program, which can be directly obtained, while the performance of MILP-CCIL is estimated by averaging many simulated episodes. The latter is seen to achieve better maximum regret than the former for low values of \(L\). Interestingly, and perhaps unexpectedly, unlike for the Twin-States domain, the performance of MILP-CCL is not monotonic in boundary \(L\). The explanation lies in the fact that though the MILP-CCL policy is a deterministic function of history, the part of the policy that occurs after the boundary \(L\) when viewed as a function of env-state alone is stochastic.

This is because the knowledge-state at time \(L\) is stochastic due to the stochastic transition dynamics (recall that the policy after \(L\) is allowed to condition on the knowledge state at time \(L\)). Thus if \(L\) is too large, the agent cannot take advantage of this stochasticity and suffers larger regret than for intermediate values of \(L\). On the other hand if \(L\) is too small, then the knowledge-state at \(L\) is not informative enough to be helpful. Also interestingly, MILP-CCIL can take advantage of this implicit stochasticity using smaller \(L\). However, when \(L\) is large, MILP-CCIL achieves the same poor performance as MILP-CCL, because when \(L\) is large the agent is likely to be in the vertical corridor where it no longer gets new knowledge about how many slippery cells there are and therefore iterative lookahead does not help.

### Conclusion

In this paper we developed a commitment semantics for achieving a specific state by a certain time with at least a certain probability in environments that have non-probabilistic uncertainty about the possible MDP the committing agent is facing as well as probabilistic uncertainty about the consequences of actions (within the true MDP). Our Commitment Constrained Lookahead (CCL) family of algorithms plan (offline) low-regret policies respecting the commitment semantics. We provided analysis and empirical results on the impact of the knowledge-state-update boundary, which is an input-parameter to CCL, on the performance of the planned policy. We extended CCL to Commitment Constrained Iterative Lookahead (CCIL), which is an iterative algorithm that adjusts the policy online. Exact CCL and CCIL require solving non-convex programs and thus we also introduced a MILP formulation that restricts the agent to deterministic policies. Our empirical results indicate that the MILP versions of both CCL and CCIL outperform baseline methods, and that CCIL is more robust than CCL.

### Acknowledgments

This work was supported in part by the Air Force Office of Scientific Research under grant FA9550-15-1-0039. Any opinions, findings, conclusions, or recommendations
expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References
Al-Saqqar, F.; Bentahar, J.; Sultan, K.; and El-Menshawy, M. 2014. On the interaction between knowledge and social commitments in multi-agent systems. *Applied Intelligence* 41(1):235–259.

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Bannazadeh, H., and Leon-Garcia, A. 2010. A distributed probabilistic commitment control algorithm for service-oriented systems. *IEEE Transactions on Network and Service Management* 7(4):204–217.

Castelfranchi, C. 1995. Commitments: From individual intentions to groups and organizations. In *Proceedings of the International Conference on Multiagent Systems*, 41–48.

Chesani, F.; Mello, P.; Montali, M.; and Torroni, P. 2013. Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems* 27(1):85–130.

Cohen, P. R., and Levesque, H. J. 1990. Intention is choice with commitment. *Artificial Intelligence* 42(2-3):213–261.

Jennings, N. R. 1993. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review* 8(3):223–250.

Mallya, A. U., and Huhns, M. N. 2003. Commitments among agents. *IEEE Internet Computing* 7(4):90–93.

Singh, M. P. 1999. An ontology for commitments in multi-agent systems. *Artificial Intelligence in the Law* 7(1):97–113.

Winikoff, M. 2006. Implementing flexible and robust agent interactions using distributed commitment machines. *Multi-agent and Grid Systems* 2(4):365–381.

Witwicki, S. J., and Durfee, E. H. 2009. Commitment-based service coordination. *International Journal of Agent-Oriented Software Engineering* 3(1):59–87.

Xing, J., and Singh, M. P. 2001. Formalization of commitment-based agent interaction. In *Proceedings of the 2001 ACM symposium on Applied computing*, 115–120. ACM.

Xuan, P., and Lesser, V. R. 1999. Incorporating uncertainty in agent commitments. In *International Workshop on Agent Theories, Architectures, and Languages*, 57–70. Springer.

Zhang, Q.; Durfee, E. H.; Singh, S. P.; Chen, A.; and Witwicki, S. J. 2016. Commitment semantics for sequential decision making under reward uncertainty. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 3315–3323.