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...
The goal of the Twin-States problem is to find a policy \( \pi \) that maximizes the expected cumulative reward under the optimal policy respecting the semantics of commitment \( c \). Let \( \Pi_c \) be the set of all history-dependent stochastic policies that satisfy Equation (2). Then, the maximin regret of commitment \( c \) is defined as:

\[
\rho^c = \max_{k \in \kappa_0} U^c(k) - \min_{\pi \in \Pi_c} U^\pi(k)
\]

where \( \rho^c \) is the maximum regret of policy \( \pi \) under the optimal policy respecting the semantics of commitment \( c \) if the true MDP is \( k \).

Finding \( \rho^c \) 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.,

\[
\rho^c = \max_{k \in \kappa_0} U^c(k) - \min_{\pi \in \Pi_c} U^\pi(k)
\]

Let \( \Pi^*_c \) be the set of policies that minimizes the maximum regret while respecting the commitment semantics,

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

Observation 1 says that in general it is not sufficient to search over policies that are optimal for some MDP.

**Observation 1.** Let $\pi^*_c(k) = \arg \max_{\pi \in \Pi_c} U^c(k)$ be a policy respecting commitment $c$ that is optimal if the true MDP is $k$. Then, in general we have $\pi^*_c(k) \notin \Pi^*_c$, $\forall k$.

Observation 2 says that in general it is not sufficient to greedily pick the next action that minimizes the maximum myopic regret.

**Observation 2.** Let $\pi_G$ be the greedy policy under which the agent selects the next action that minimizes the maximum myopic regret over the possible MDPs consistent with the current knowledge, i.e.

$$a_t = \arg \min \max_{k \subseteq \kappa_t} \{ \max_{a' \in a} [R_{s,a'}(k) - R_{s,a}(k)] \}.$$  

Then, in general we have $\pi_G \notin \Pi^*_c$.

Observation 3 says that it is possible that no policy in $\Pi^*_c$ is deterministic even if all MDPs in the environment are deterministic.

**Observation 3.** There exists an environment $\mathcal{E}$ where all MDPs are deterministic, i.e., $\forall k, s, a \exists s'$ such that $P^c_{s,a}(k) = 1$, and no policy in $\Pi^*_c$ 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 $\pi \in \Pi_c$ such that if the agent executes policy $\pi$ for the first $t > 0$ time steps starting in state $s_0$, the history generated, $h_t$, is such that

$$\forall \pi', \exists k \in \kappa_t \Pr_{\pi'}(S_T \in \Phi|S_t = s_t; k) < p.$$  

**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 $\Pi_0$ be the set of all Markov policies, i.e., policies that choose actions solely as a function of the current env-state (and ignore $\kappa$). Assuming $\Pi_0 \cap \Pi_c = \emptyset$, our Commitment Constrained No-Lookahead (CCNL) method of Figure 3 finds a minimax regret Markov policy respecting the

Figure 2: Starting in state A, the agent commits to reaching the absorbing state D at time step two with at least probability .8. If the agent happens to be in state C at time step one, there is no plan that reaches state D from state C with probability at least .8 (verifying Observation 4). There are two possible reward functions $R_1$ and $R_2$ shown above. Even though re-planning from state C does not yield a plan that leads to state D with probability 0.8, the new plan will nonetheless reduce regret because at time step 1 we will know which reward function applies and can therefore choose the more rewarding action in state C.

\[
\min \max_k U^*_c(k) - U(k) \quad (3a)
\]

subject to

\[
\forall k \quad U(k) = \sum_{s,a} x_{sa}(k) R_{sa}(k) \quad (3b)
\]

\[
\forall k, s, a \quad x_{sa}(k) \geq 0 \quad (3c)
\]

\[
\forall k, s', a \quad x_{sa}(k) = \sum_{s,a} x_{sa}(k) P^c_{s,a}(k) + \delta_{s',s_0} \quad (3d)
\]

\[
\forall k, k', s, a \quad \frac{x_{sa}(k)}{x_{sa}(k')} = \frac{x_{sa}(k')}{x_{sa}(k)} \quad (3e)
\]

\[
\forall k \quad \sum_{s' \in \Phi} \sum_a x_{sa}(k) \geq p \quad (3f)
\]

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 $s_0$ and the transition function of the $k^{th}$ MDP is $P(k)$, where $\delta_{s',s_0}$ is the Kronecker delta that returns 1 when $s' = s_0$ 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:
\[
\min_{\pi \in \Pi_c \cap \Pi_L} \rho_c^\pi.
\] (4)

For MDP \(k\), each policy \(\pi\) has a corresponding occupancy measure \(x^\pi(k)\) for env-state-action pairs:
\[
x^\pi_{sa}(k) := E_x \left[ \sum_{t=0}^{T-1} 1\{s_t = s, a_t = a\} | S_0 = s_0; k \right].
\]

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
\[
\pi(a|s) = \frac{x_{sa}(k)}{\sum_{a'} x_{sa'}(k)}.
\] (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:
\[
\pi(a|h_t) = \begin{cases} 
\pi(a|h_t) & t < L \\
\pi(a|s_t, b_L) & t \geq L,
\end{cases}
\]

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:
\[
\min_{\pi \in \Pi_c \cap \Pi_L} \rho_c^\pi.
\] (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

\[
\pi_L(a|h_t) = \begin{cases} 
\pi_L(a|h_t) & t < L \\
\pi_L(a|s_t, b_L) & t \geq L,
\end{cases}
\] (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
\[
\rho_c^\pi_L \leq \rho_c^{\pi_0^*}.
\]
\[
\begin{align*}
\min_{x,y} & \max_{k \in \kappa_0} \quad U^*_c(k) - U(k) \\
\text{subject to} & \\
\forall k \in \kappa_0 & \\
U(k) = & \sum_{b \in B^0_{[0,L]}, a} y_{ba}(k) \tilde{R}_{ba}(k) + \sum_{b,s,a \in B^0_{[0,L]}, s,a} x_{sa}^b(k) R_{sa}(k); \\
\forall k, b, a & \quad y_{ba}(k) \geq 0; \\
\forall k, b' = (s', k') \in B^0_{[0,L]} & \quad \sum_{a'} y_{ba'}(k) = \sum_{a'} y_{ba'}(k') \quad \text{(utility if MDP } k \text{ is true)}; \\
\forall k, b, a & \quad y_{ba}(k) \geq 0; \\
\forall k, b, l \in B^0_{[0,L]}, a & \quad \frac{\sum_{a'} y_{ba'}(k)}{\sum_{a'} y_{ba'}(k')} = \frac{y_{ba}(k)}{y_{ba}(k')} \quad \text{(policies via } y(k) \text{ and } y(k') \text{ are consistent)}; \\
\forall k, b, l \in B^0_{[0,L]}, a & \quad \frac{\sum_{a'} y_{ba'}(k)}{\sum_{a'} y_{ba'}(k')} = \frac{y_{ba}(k)}{y_{ba}(k')} \quad \text{(define } y_{ba} \text{ as the prob of reaching } b_L) \quad \text{(7f)}; \\
\forall k, b, l \in B^0_{[0,L]}, a & \quad \frac{\sum_{a'} y_{ba'}(k)}{\sum_{a'} y_{ba'}(k')} = \frac{y_{ba}(k)}{y_{ba}(k')} \quad \text{(define } y_{ba} \text{ as the prob of reaching } b_L) \quad \text{(7f)}; \\
\forall k, b, l \in B^0_{[0,L]}, s, a & \quad \sum_{a'} x_{s'a}^b(k) = \sum_{a'} x_{s'a}^b(k') \quad \text{(policies via } x(k) \text{ and } x(k') \text{ are consistent)}; \\
\forall k, b, l \in B^0_{[0,L]}, s, a & \quad \sum_{a'} x_{s'a}^b(k) = \sum_{a'} x_{s'a}^b(k') \quad \text{(commitment semantics)} \quad \text{(7j)}; \\
\forall b \in B^0_{[0,L]}, k, k', s & \quad x_{sa}^b(k) = \sum_{a'} x_{s'a}^b(k') \quad \text{(7i)}; \\
\forall b \in B^0_{[0,L]}, k, k', s & \quad x_{sa}^b(k) = \sum_{a'} x_{s'a}^b(k') \quad \text{(7i)}; \\
\forall k \in \kappa_0 & \quad \sum_{b \in B^0_{[0,L]}, s \in \Phi, a} x_{sa}^b(k) \geq p \quad \text{(commitment semantics)} \quad \text{(7j)}; \\
\end{align*}
\]

Figure 5: CCL program. To derive this program, we first define the knowledge state-based transition function \(\tilde{P}_{ba}^b(k) = 1_{\{k \in \kappa\}} \Pr(b'|b, a; k)\), and \(\tilde{R}_{ba}(k) = 1_{\{k \in \kappa\}} R_{sa}(k)\), where \(\tilde{P}_{ba}^b(k)\) is the probability that the next knowledge state is \(b'\) upon taking action \(a\) in knowledge state \(b\), given that the true MDP is \(k\). Similarly, \(\tilde{R}_{ba}(k)\) is the reward of doing action \(a\) in knowledge state \(b\) given that the true MDP is \(k\). Note that if MDP \(k\) is ruled out according to knowledge state \(b\), then we define \(\tilde{P}_{ba}^b(k) = 0\) and \(\tilde{R}_{ba}(k) = 0\). Given policy \(\pi\), one can use \(\tilde{P}(k)\) to calculate the corresponding occupancy measure \(y^\pi(k)\) for knowledge state-action pairs as follows: \(y^\pi_{ba}(k) := E_{\pi} \left[ \sum_{t=0}^{T-1} 1_{(b_{t+1} = b, a_t = a)} \right] \). We use \(B^0_{[0, L]}\) to denote the set of reachable knowledge states after executing exactly \(L\) actions from knowledge state \(b\), and \(B^0_{[l_1, l_2]} = \bigcup_{l=1}^{l_2} B^0_{[l_1, l_2]}\) to denote the set of reachable knowledge states from \(b\) by executing any \(l\) actions such that \(l \in [l_1, l_2]\). Because time is a state feature, \(B^0_{[0, L]}\) and \(B^0_{[l_1, l_2]}\) are disjoint if \(l \neq l'\). CCL generates beforehand all reachable knowledge states from initial knowledge state \(b_0\) within \(L\) actions, \(B^0_{[0, L]}\). The state-action measures also enable us to express the expected cumulative reward conveniently in constraint (7b) where the first RHS term sums up the reward of the first \(L\) time steps and the second term the remaining \(T - L\) time steps. The state-action measures also enable us to express commitment semantics conveniently via constraint (7j). Constraints (7c), (7d), and (7e) on \(y\) are the counterparts of (3c), (3d), and (3e) in Figure 3. Similarly, constraints (7g), (7h), and (7i) are the counterparts for \(x\).

However, one has to be careful in using deeper boundaries because the performance of CCL is guaranteed to be monotonically non-decreasing in \(L\) only when transition dynamics is invariant across MDPs, but this monotonicity cannot be guaranteed in general, as stated in Theorem 3 and Theorem 4.

**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 
\[
\rho_c^{\pi^{L'}} > \rho_c^{\pi^{L}},
\]
where \(\pi^{L'}_c\) and \(\pi^{L}_c\) 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

$$\rho_{c,b}^{\pi_{L'}^{b}} \leq \rho_{c,b}^{\pi_{L}^{b}},$$

where $\pi_{L}^{b}$ and $\pi_{L'}^{b}$ 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 $\pi_{L}^{b}$ derived by solving the program in Figure 5 and ends up in knowledge state $b_L \in B^K_{\text{no}}$. Instead of executing the remaining $T-L$ actions prescribed by $\pi_{L}^{b}$, 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 $k \in \kappa_L$, update the commitment probability as the achieved probability if the agent were to stick with $\pi_{L}^{k}$ from $s_L$:

$$p(k) = \Pr_{\pi_{L}^{k}}(S_T \in \Phi | S_L = s_L; k). \quad (9)$$

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:

$$\max_{\pi} \left[ \frac{1}{T} \sum_{t=1}^{T-1} R_{\mathcal{S}_t \mathcal{A}_t}(k) | S_L = s_L; k \right] \quad (10)$$

subject to $\Pr(\mathcal{T} \in \Phi | S_L = s_L; k) \geq p(k)$

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 $\pi_{L}^{b}$ itself is a solution. CCIL 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 $\pi_{L}^{\text{ccil}}$ be the history-dependent policy defined as Algorithm 1. We have $\pi_{L}^{\text{ccil}} \in \Pi_c$, 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 $a_t \sim \pi_i(\cdot | b_t)$ 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.       $\pi_{t+1} \leftarrow \pi_i$;
9.       $t \leftarrow t + 1$;
10. end
11. for $k \in \kappa_t$ do
12.    $p(k) \leftarrow \Pr_{\pi_k}(S_T \in \Phi | S_t = s_t; k)$;
13.    $U_{s_t,p(k)}^{*}(k) \leftarrow$ 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 $U_{s_t,p(k)}^{*}(k)$;
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 at 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
and compare it with that found using the MILP formulation.

Here we set the time horizon to two so that Short horizon.

We evaluate the performance of CCL and CCIL, under var-
we can find an exact stochastic minimax regret CCL policy
Here we set the time horizon to two so that

boundaries against simple policy construction methods.

compare the performance of CCL and CCIL using various
evaluate the loss of the MILP formulation in a domain where
we investigate aleatoric uncertainty. For this domain, we let

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. $\pi_k^* = \arg\max_{\pi \in \Pi_t} U^\pi(k)$. The MDPs-Best policy is the one out
  of $\{\pi_k^*\}_{k=1}^2$ 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.

$$a_t = \arg\min_{a \in A_t} \max_{k \in \kappa_t} \max_{a' \in A_t} \{R_{s_t,a}(k) - R_{s_t,a'}(k)\}$$

where $A_t$ is the set of actions available at time $t$ that are
chosen to guarantee the commitment semantics is respec-
ted. 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.


Figure 6: Maximum regret in the Twin-States domain of ex-
act CCL, MILP-CCL, and MILP-CCIL when horizon is two.
Markers “o” for MILP-CCIL overlap with makers “x” for
MILP-CCL when $L = 1, 2$. (Note that MILP-CCIL is not defined for $L = 0$).

**Empirical Results**

We evaluate the performance of CCL and CCIL, under var-
ious 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 pro-
grams 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

![Graph showing maximum regret in the Twin-States domain of exact CCL, MILP-CCL, and MILP-CCIL.](image)

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

\[
\pi_L(a|h_t) = \begin{cases} 
\pi_L(a|h_t) = 1_{\{\Delta b_{a-1}\}} & t < L \\
\pi_L(a|s_t, b_L) = 1_{\{\Delta b_{a-1}\}} & t \geq L.
\end{cases} \tag{11}
\]

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

$$\min_{x, z} z$$

and add the following constraints on $z$

$$\forall k \in \kappa_0, z \geq U_c^*(k) - U(k)$$

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

**Empirical Results**

We evaluate the performance of CCL and CCIL, under var-
ious 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 pro-
grams 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
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 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).

Table 1: Max regret with varying horizon in Twin-States.

<table>
<thead>
<tr>
<th>Horizon $T$</th>
<th>Greedy</th>
<th>MDPs-Best</th>
<th>MILP-CCL, $L = 0$</th>
<th>MILP-CCL, $L = 1$</th>
<th>MILP-CCIL, $L \in [3, T]$</th>
<th>MILP-CCIL, $L = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>13</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>19</td>
<td>15</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>25</td>
<td>19</td>
<td>9</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>21</td>
<td>31</td>
<td>22</td>
<td>11</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

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


Technical Proofs

Proof of Theorem 1. We need to show any policy in $\Pi_c \cap \Pi_L$ one-to-one maps to a feasible solution to the program in Figure 5.

For any policy $\pi \in \Pi_c \cap \Pi_L$, we are going to define vectors $m(\pi)$ and $n(\pi)$ such that they satisfy the constraints of the program in Figure 5 if treated as the decision variables $x$ and $y$, respectively. Given any policy $\pi \in \Pi_c \cap \Pi_L$, let $n(\pi,k)$ be its knowledge state-action occupancy measure if the true MDP is $k$ for knowledge states in $B_{[0,L]}^b$, and $m(\pi,k)$ be its env-state-action occupancy measure for env-states from time step $L$ on.

$$\forall b \in B_{[0,L]}^b, \ a \n b_{ba}(\pi,k) = Pr(B_t = b, A_t = a, B_L = b, 0; k)$$

$$\forall s, a \n b_{ba}^{ba}(\pi,k, a) = \begin{cases} Pr_\pi(S_t = s, A_t = a, B_L = b, 0; k) & t \geq L \\ 0 & t < L \end{cases}$$

where $t$ is the time of knowledge state $b$. We next show $n_{ba}(\pi,k)$ satisfies the constraints if treated as $y_{ba}(k)$ and $m_{ba}^{ba}(\pi,k)$ satisfies the constraints if treated as $x_{ba}^{ba}(k)$.

If treated as $y_{ba}(k)$, $b_{ba}(\pi,k)$ satisfies constraint (7c) because $n_{ba}(\pi,k) \geq 0$.

If treated as $y_{ba}(k)$, $b_{ba}(\pi,k)$ satisfies constraint (7d) because if $b' = 0$, for the LHS of (7d) we have

$$\sum_{a'} b_{ba}(\pi,k) = \sum_{a'} b_{ba}(\pi,k) (because b' = 0)$$

$$= \sum_{a'} Pr(B_0 = b, A_0 = a | B_0 = b; k)$$

$$= Pr(B_0 = b)$$

and for the RHS of (7d) we have

RHS of (7d)

$$= \sum_{b,a} n_{ba}(\pi,k) P_{ba}^{ba}(k) + \delta_{b_0,b_0} \mathbb{1}_{k \in \kappa'}$$

$$= \sum_{b,a} n_{ba}(\pi,k) P_{ba}^{ba}(k) + \delta_{b_0,b_0} \mathbb{1}_{k \in \kappa} (because b' = 0)$$

$$= 0 + 1 = LHS \text{ of } (7d)$$

If $b' \in B_{[0,L]}^b \setminus \{b_0\}$, for the LHS of (7d) we have

$$\sum_{a'} b_{ba}(\pi,k) = \sum_{a'} Pr(B_t = b', A_t = a' | B_0 = b; k) (t' \text{ is time of } b')$$

$$= Pr(B_t = b')$$

and for the RHS of (7d) we have

$$RHS \text{ of } (7d)$$

$$= \sum_{b,a} n_{ba}(\pi,k, k) P_{ba}^{ba}(k) + \delta_{b_0,b_0} \mathbb{1}_{k \in \kappa'}$$

$$= \sum_{b,a} n_{ba}(\pi,k, k) P_{ba}^{ba}(k) + \delta_{b_0,b_0} \mathbb{1}_{k \in \kappa} (because b' = 0)$$

$$= 0 + 1 = LHS \text{ of } (7d)$$

Thus, we know that $\pi_0 \in \Pi_L$.

We first prove Theorem 4 and then Theorem 3.
Proof of Theorem 4. It's sufficient to show that the statement holds when $L' = L + 1$. We next show that given any policy $\pi_L \in \Pi_L$, there exists an $(L + 1)$-updates policy, $\pi_{L+1}$, that mimics $\pi_L$ when $P(k) = P(k') \forall k, k'$, and therefore $\rho_{\pi^{L+1}_c} \leq \rho_{\pi^{L}_c}$.

For the first $L$ actions, an $(L + 1)$-updates policy can map the current knowledge state to a distribution of the next actions identical to $\pi_L$. The action that is going to take at time step $L$ by $\pi_L$ can also be recovered by an $(L + 1)$-updates policy, which gives

$$\pi_{L+1}(a|b_t) = \begin{cases} \pi_{L+1}(a|b_t) = \pi_L(a|b_t) & t < L \\ \pi_{L+1}(a|b_L) = \pi_L(a|s_L, b_L) & t = L \end{cases}$$

If $P(k) = P(k') \forall k, k'$, then $\pi_{L+1}$ can also recover $\pi_L$ for $t \geq L + 1$. To see this, note if $P(k) = P(k') \forall k, k'$ we have

$$\Pr(b_t|b_{L+1}; k) = \Pr(b_t|b_{L+1}; k'), \forall \pi_L, k, k' \in \kappa_{L+1}$$

Therefore, under any $L$-updates policy $\pi_L$ and conditioned on being in knowledge state $b_{L+1}$ at time step $L + 1$, the agent thereafter selects actions according to $\pi_L(\cdot|s_t, b_L)$ with probability defined in Equation (12) that does not depend on $k$. Because the transition function is known, the occupancy measure of $\pi_L$ for $t \geq L + 1$ conditioned on any $b_{L+1}$ can be achieved by a stochastic Markov policy from $b_{L+1}$, which can be expressed by an $(L + 1)$-updates policy as $\pi_{L+1}(\cdot|s_t, b_{L+1})$.

Proof of Theorem 3. In general $P(k) = P(k') \forall k, k'$ does not hold, and therefore condition (12) in the proof of Theorem 4 does not hold. Let $b_{L+1} = (s_{L+1}, \kappa_{L+1})$ be a knowledge state at time step $L + 1$.

Inspired by this we now give an example as a formal constructive proof. In this example the env-state space is $\{0, 1, 2, 3\}$ and let state 0 be the initial state. There are two actions $a_0, a_1$ and $K = 2$ possible MDPs. The time horizon is three and the commitment probability is zero. The transition dynamics and reward of these MDPs are as follows.

1. In MDP $k = 1$, $\Pr(1|0, a; k = 1) = 0.9$, $\Pr(2|0, a; k = 1) = 0.1$, $\Pr(3|1, a; k = 1) = \Pr(3|2, a; k = 1) = 1.0$ for both $a = a_0, a_1$. State 3 is an absorbing state. Doing $a_0$ in state 3 gives a positive unit of reward. There’s no reward elsewhere.

2. In MDP $k = 2$, $\Pr(1|0, a; k = 1) = 0.1$, $\Pr(2|0, a; k = 1) = 0.9$, $\Pr(3|1, a; k = 1) = \Pr(3|2, a; k = 1) = 1.0$ for both $a = a_0, a_1$. State 3 is an absorbing state. Doing $a_1$ in state 3 gives a positive unit of reward. There’s no reward elsewhere.

The maximum regret when $L = 2$ is 0.5, but $L = 1$ can achieve a maximum regret of 0.1.

Proof of Theorem 5. We need to show

$$\Pr_{\pi^{\text{L}}_c}(S_T \in \Phi|S_0 = s_0; k) \geq \forall k \in \kappa_0.$$

Let $\pi_L$ be the CCL $L$-updates policy derived from the program in Figure 5. For any $k \in \kappa_0$, we can calculate the achieved commitment probability of $\pi^{\text{L}}_c$ by conditioning on the knowledge state it will visit at time $L > 0$,

$$\Pr_{\pi^{\text{L}}_c}(S_T \in \Phi|S_0 = s_0; k) = \sum_{b_L} \Pr_{\pi^{\text{L}}_c}(B_L = b_L|S_0 = s_0; k) \Pr_{\pi^{\text{L}}_c}(S_T \in \Phi|B_L = b_L; k) = \sum_{b_L} \Pr_{\pi^{\text{L}}_c}(B_L = b_L|S_0 = s_0; k) \Pr_{\pi^{\text{L}}_c}(S_T \in \Phi|B_L = b_L; k) \geq \sum_{b_L} \Pr_{\pi^{\text{L}}_c}(B_L = b_L|S_0 = s_0; k) \Pr_{\pi^{\text{L}}_c}(S_T \in \Phi|B_L = b_L; k) = \Pr_{\pi^{\text{L}}_c}(S_T \in \Phi|S_0 = s_0; k) \geq p$$

The second equality holds because $\pi^{\text{L}}_c$ is identical to $\pi_L$ in the first $L$ steps. The first inequality holds because CCIL iteratively applies $L$-step lookahead in Algorithm 1 line 15 with the commitment probability achieved by the policy of the previous iteration calculated in line 12. The modified program in Algorithm 1 line 15 is always feasible because $\Pi_0 \subset \Pi_L$ which is shown in the proof of Theorem 2.

\[\square\]