Autonomous Agents and Multi-Agent Systems (2020) 34:19
https://doi.org/10.1007/s10458-020-09443-0

Semantics and algorithms for trustworthy commitment achievement under model uncertainty

Qi Zhang[1] · **Edmund H. Durfee**[1] · **Satinder Singh**[1]

Published online: 18 January 2020

Abstract
We focus on how an agent can exercise autonomy while still dependably fulfilling commitments it has made to another, despite uncertainty about outcomes of its actions and how its own objectives might evolve. Our formal semantics treats a probabilistic commitment as constraints on the actions an autonomous agent can take, rather than as promises about states of the environment it will achieve. We have developed a family of commitment-constrained (iterative) lookahead algorithms that provably respect the semantics, and that support different tradeoffs between computation and plan quality. Our empirical results confirm that our algorithms' ability to balance (selfish) autonomy and (unselfish) dependability outperforms optimizing either alone, that our algorithms can effectively handle uncertainty about both what actions do and which states are rewarding, and that our algorithms can solve more computationally-demanding problems through judicious parameter choices for how far our algorithms should lookahead and how often they should iterate.

Keywords Commitment semantics · Model uncertainty · Constrained planning · Sequential decision making

1 Introduction

To pursue its assigned objectives in the face of dynamic and unpredictable circumstances, an agent needs the autonomy to flexibly adopt different actions as needed. But such autonomy risks making the agent more unpredictable to other agents it is cooperating with. Specifically, to be trustworthy, the agent's exercise of its autonomy must avoid violating social commitments that it has made to others. This balance, between satisfying others' expectations by employing autonomy to improve individual effectiveness, versus satisfying others'

✉ Qi Zhang
qizhg@umich.edu

Edmund H. Durfee
durfee@umich.edu

Satinder Singh
baveja@umich.edu

[1] Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA

expectations by tempering autonomy to adhere to social commitments, is the focus of this article.

Most multiagent systems research into social commitments, and trust in agents to meet them, investigates constructs and protocols by which agents can express commitments, test for commitment satisfaction, and develop models of reputation and trust. Through such mechanisms, self-interested agents (or their developers) gain incentive for meeting commitments, as finding gullible agents to exploit by reneging on commitments becomes harder. Such a perspective assumes that agents (or their developers) know how to satisfy commitments and earn the trust of others, and need to be incentivized to do so. This perspective is sensible, for example, in transactional contexts, where goods, services, and money are exchanged.

Our emphasis in this article is different. We assume that agents are cooperative, so incentivizing is not the issue. Instead, we do *not* assume that it is clearcut how *autonomous* agents should satisfy commitments in the inherently uncertain contexts where autonomy is needed. An example in human systems is the commitment a surgeon makes when treating a patient, where the motives of the surgeon and patient are aligned, but the planned surgical procedure might not work, or the surgeon might discover during the surgery an overriding problem to address instead. The patient is counting on the surgeon to autonomously respond to emergent circumstances, within the bounds of standards of practice/care. Even though the outcome of the surgery might not be what everyone expected, the surgeon still could be deemed as to have merited the trust of the patient in fulfilling the commitment made.

For artificial autonomous agents, we seek to distill insights from such human settings into computationally operational terms, by defining a clear semantics for what it means to achieve a commitment despite inherent uncertainty, and by formulating decision-making algorithms for an agent that provably adhere to those semantics. This article provides one set of answers to these questions, though others might be possible. The key ideas behind our answers include:

- A recognition that, in uncertain settings, commitments can only be made to what actions an agent will attempt, rather than to what outcomes will result from its actions.
- The importance of committing to acting within a *space* of possible actions/plans, rather than to a specific single action/plan, so an agent can retain latitude for exercising autonomy.
- The ability to succinctly characterize a space of possible actions/plans in terms of a probabilistic commitment to reach one of a set of possible states.
- A family of algorithms for provably and flexibly achieving probabilistic commitments that allow different tradeoffs between computational effort exerted and the optimality of autonomous behavior.

We described this general area of research and our early intuitions for solving it in a workshop paper that was later included in a published collection of best workshop papers [11]. Select technical details have appeared in an unpublished workshop paper [42]. In a conference paper [43], we proposed similar commitment semantics, under Bayesian reward uncertainty, to those in this article, but described solution techniques that included a less robust iterative method that can violate the prescriptive commitment semantics in subtle ways. The improved iterative method in this article (Sect. 4.4) supersedes the previous flawed one, providing provable guarantees on fulfilling the prescriptive commitment semantics. Versions of the improved methods in this article, but applied to non-Bayesian

setting, have appeared in a conference paper [44]. In contrast, this article goes into detail about (previously unpublished) Bayesian versions of the improved solution methods, formally proving their adherence to our semantics. It also collects together content that had previously appeared only in non-archival forms. Finally, beyond any of our work that has appeared elsewhere, this article additionally provides more details and intuitions, and a more extensive empirical evaluation.

In the sections that follow, after briefly summarizing past work on commitments in agent systems (Sect. 2), we describe the decision-theoretic foundations of our work and the formal semantics of a probabilistic commitment (Sect. 3). From there, we describe a family of decision-making algorithms that make different computation/optimality tradeoffs, with theoretical results proving their adherence to our commitment semantics (Sect. 4), along with strategies to solve the problems more efficiently (Sect. 5). Our empirical results highlight the effectiveness of our principled approach and the tradeoffs among the different algorithms (Sect. 6). Finally, the article finishes with a summary of the contributions, and of directions for future work (Sect. 7).

2 Related work

A comprehensive overview of research into using computational methods to characterize and operationalize social commitments in terms of formal (temporal and modal) logic has appeared [32], and is based on literature in this field (e.g., [2, 5–7, 19, 31]). These formulations support important objectives like the provable pursuit of mutually agreed-upon goals, and codifying conventions and protocols for managing uncertainty (e.g., [16, 38, 40]). As an example of a convention, an agent that determines that it will not keep a commitment might be obligated to inform dependent agents [16].

There has been substantial work in the field on formal methods for developing protocols, with provable properties, for agents who are modeling and communicating about commitments. The focus is on the lifecycle of a commitment, from its initial proposed creation, to the mutual agreement to adopt it, to determining whether it has been fulfilled, to whether it is time to abandon it. Over the lifecycle, it is important that interacting agents engage in a communication protocol that ensures their beliefs about the status of a shared commitment are aligned. Günay et al. [12, 13] have developed a formal language that provides for non-deterministic elements in agents' beliefs, along with a model-checking algorithm for analyzing agents' compliance with commitments. Sultan et al. [35] develop another model-checking technique for verifying social commitments specified using a modal logical language for uncertain settings. To improve alignment when an agent suspects another might not be conscientious about providing updates, Pereira et al. [24] have developed an approach where an agent can use observations of another's actions to infer when that other agent has abandoned its commitment, based on if its current actions do not fit any (known) plan to fulfill the commitment. Their work assumes a commitment semantics that requires an agent who has adopted a commitment to singlemindedly pursue that commitment. In general, this semantics is overly restrictive, because an agent might need to interleave actions in the pursuit of multiple commitments, along with its own local goals, at any given time.

The above acknowledges that the agents might not achieve outcomes they have committed to, and focuses on improving awareness across agents as to the status of all their shared commitments. But if agents can arbitrarily and unilaterally decide to drop a commitment,

the commitment loses predictive value for coordination between the agents. Some of the logical formulations above (e.g., [16]) enumerate conditions where an agent is allowed to abandon its local component of a mutual goal, where in general these conditions are either: (1) when the agent believes it is impossible to achieve its local component; (2) when the agent believes the mutual goal is not worth pursuing anymore; or (3) when the agent believes one or more of the other agents participating in the mutual goal have abandoned their local components of it. These conditions are logically reasonable, but fail to impose a prescriptive semantics for the agent to use in making local decisions. For example, to satisfy the first condition, is an agent never allowed to take an action that has even a small chance of rendering its local component unachievable? What if all of its actions have such a chance? For the second condition, if an agent can unilaterally drop a commitment whenever its preferred goal changes, then has it really committed in the first place?

To make an agent more predictable, a commitment can be paired with conditions under which it is guaranteed to hold [1, 27, 32, 37]. In transactional settings, for example, an agent could commit to providing a good or service on the condition that it first receives payment. However, if conditions can be over anything, then they can make commitments worthless because a commitment might be conditioned simply on no better option coming along. Sandholm and Lesser [28] recognized the general impracticality of enumerating all the conditions that might affect commitment adherence, and, even if the conditions could be specified, in verifying they hold in a distributed setting. Their solution was a contracting framework where a decommitment penalty is associated with each commitment, so as to accommodate uncertainty but discourage frivolous decommitment. However, even though the **recipient** of a commitment will know it will be compensated if the commitment is abandoned, it in general will be unable to know how likely that will be, since it cannot look inside the **provider** of the commitment to discern how likely it is that its actions to achieve the commitment will fail, or that it will decide that other goals should take priority.

Therefore, an alternative to a decommitment penalty is for the commitment provider to summarize the likelihood that its commitment's various conditions will jointly hold (e.g., a factory's suppliers will meet deadlines, its workers will not strike, its shippers will fulfill orders, etc.) into a summary probability. Hence, a probabilistic commitment [4, 39, 41] is a form of conditional commitment where the details of the conditions have been replaced by an estimate of the probability that they will hold. Xuan and Lesser [41] have explained how probabilistic commitments improve joint planning by allowing agents to find policies that are responsive to possible contingencies, including even unlikely ones, and computing appropriate alternative courses of action as the probabilities for commitments being met change. A more myopic (to be more tractable) variation of this approach was developed for the DARPA Coordinators program [18], where instead of anticipating ways that probabilities might change, the recipient would revise its plans only when the commitment provider would send an updated probability of the commitment being satisfied. These prior approaches however only treat commitment probabilities as predictions about how the provider's plan will affect recipients. In contrast, our goal is that probabilistic commitments not only provide such predictive information to the recipient, but also impose prescriptive semantics on the provider to influence its behavior into a good faith effort towards making those predictions come true.

Our work, summarized in this article, is the first to develop prescriptive commitment semantics under decision-theoretic model uncertainty, along with algorithms that operationalize this semantics for faithful commitment pursuit. The model uncertainty that we consider is a form of partial observability, and thus the algorithms we develop can be viewed as extensions of existing techniques for solving (unconstrained) partially-observable

Markov decision problems [15, 17, 33]. Our commitment semantics prescribes additional constraints to the original planning problem, and we develop algorithms that exactly meet the commitment constraints under partial observability. Existing work has developed methods for constrained decision-theoretic planning without model uncertainty [3], or has solved the constraints only approximately [29, 25]. Others have also developed planning approaches for given commitments formulated using formal logic, which mainly rely on techniques of heuristic search (e.g., [21, 22, 36]). These approaches usually amount to enumerating courses of action in search for conditions that ensure the feasibility of the commitments. For example, Meneguzzi et al. [21] develop a depth-first search algorithm to generate realizable enactments of the commitment. These logic-based planning techniques deal with the provider's uncertainty about the outcomes of its actions, while we also consider the provider's uncertainty over the rewards and dynamics of its environment.

3 Problem formulation

3.1 Markov decision processes

We first provide background on Markov Decision Processes (MDPs), which form the basis of the decision-theoretic setting we adopt for the commitment provider. An MDP is formally defined by the tuple $M = (\mathcal{S}, \mathcal{A}, P, R, s_0, H)$ where $\mathcal{S}$ is the finite state space, $\mathcal{A}$ is the finite action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ ($\Delta(\mathcal{S})$ denotes the set of all probability distributions over $\mathcal{S}$) is the transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, H is the finite horizon, and s_0 is the initial state. The state space is partitioned into disjoint sets by the time step, $\mathcal{S} = \bigcup_{t=0}^{H} \mathcal{S}_t$. By taking action $a_t \in \mathcal{A}$ in state $s_t \in \mathcal{S}_t$, the environment generates a reward $r_{t+1} = R(s_t, a_t)$ and transits to a new state $s_{t+1} \in \mathcal{S}_{t+1}$ according to transition function P, i.e. $s_{t+1} \sim P(\cdot|s_t, a_t)$ meaning that s_{t+1} is stochastically drawn from the transition function. Given a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ and starting in the initial state, a random sequence of transitions $\langle s_0, a_0, r_1, s_1, \ldots, s_{H-1}, a_{H-1}, r_H, s_H \rangle$ is generated, which records the entire history up to the time horizon. The value function of π is $V^{\pi}(s) = E[\sum_{t'=t+1}^{H} r_{t'} | \pi, s_t = s]$ where t is such that $s \in \mathcal{S}_t$. The optimal policy for M, denoted as π^*, maximizes V^{π} for all $s \in \mathcal{S}$.

We give an illustrative example of an MDP shown in Fig. 1. There are four locations for each time step, labeled as A, B, C, and D. The time horizon is $H = 10$. Thus, the state space consists of 44 states, and we let $\mathcal{S}_t = \{\mathrm{A}_t, \mathrm{B}_t, \mathrm{C}_t, \mathrm{D}_t\}$. The initial state is A_0. There are two actions, α and β, and the transition function is shown in the annotations of the edges in Fig. 1. Taking any action in D yields a reward of + 10, and taking action α in the other three locations yields a reward of − 1. The rewards of all other state-action pairs are zero. We now describe the optimal policy for this MDP. In locations A, C, and D, the transition probabilities of both actions are identical, and therefore the optimal policy takes action β since it yields reward no smaller than action α. In location B, if it is not the final action then the optimal policy takes action α to transit to D which yields reward (+ 10) that outweighs the negative reward of taking action α (− 1); otherwise, the optimal policy takes action β in B as the final action.

There are several methods for solving an MDP for its optimal policy. We here summarize one based on tabular linear programming (LP) [26], which is also the computational strategy we adopt for developing algorithms for the provider. For MDP M, each

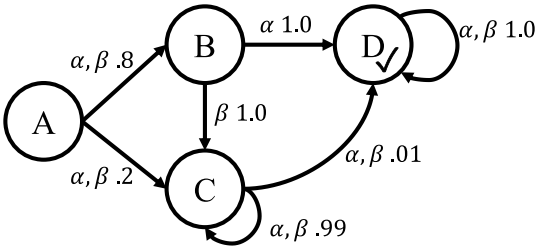


Fig. 1 There are four locations for each time step labeled as A, B, C, and D, with A being the initial location. The time horizon is $H = 10$. Thus, the state space consists of 44 states. There are two actions, α and β, and the transition function is shown in the annotations of the edges. Taking any action in D yields a reward of + 10, and taking action α in the other three locations yields a reward of − 1. The rewards of all other state-action pairs are zero

$$\begin{aligned} \max_{x} \quad & \sum\nolimits_{s,a} x(s,a)R(s,a) && \text{(1a)} \\ \text{subject to} \quad & \forall s,a \quad x(s,a) \geq 0; && \text{(1b)} \\ & \forall s' \quad \sum\nolimits_{a'} x(s',a') = \sum\nolimits_{s,a} x(s,a)P(s'|s,a) + \delta(s',s_0) && \text{(1c)} \end{aligned}$$

Fig. 2 The linear program for solving an MDP M

policy π has a corresponding occupancy measure x^π for the expected number of times action a will be taken in state s over the time horizon H, starting in initial state s_0:

$$x^\pi(s,a) = E\left[\sum_{t=0}^{H-1} 1_{\{s_t=s,a_t=a\}} | s_0, \pi\right],$$

where 1_E is the indicator function that takes value one if event E occurs and zero otherwise. We will use shorthand notation x in place of x^π when policy π is clear from the context. Policy π can be recovered from its occupancy measure via

$$\pi(a|s) = \frac{x(s,a)}{\sum_{a'} x(s,a')}.$$

Figure 2 is the linear program that solves an MDP M. It introduces the occupancy measure as decision variables, and the policy is constructed from the program's optimal solution. Constraints (1b) and (1c) guarantee that x is a valid occupancy measure, where $\delta(s', s_0)$ is the Kronecker delta that returns 1 when $s' = s_0$ and 0 otherwise. The expected cumulative reward can be expressed using x in the objective function (1a).

3.2 The decision-theoretic setting

We consider the setting in which the provider's true sequential decision-making problem is one out of K possible MDPs drawn from a known prior distribution, where all MDPs share identical state and action spaces but possibly different transition and reward functions, and the state and the reward are fully observable during execution. Formally, the environment is defined by the tuple $\mathcal{E} = \langle \mathcal{S}, \mathcal{A}, \{P_k, R_k\}_{k=1}^K, s_0, \mu_0, H \rangle$. We assume that the state space $\mathcal{S}$, the action space $\mathcal{A}$, and the time horizon H are finite. Although our commitment semantics can be straightforwardly extended beyond this assumption, the algorithms that we develop in this article to operationalize the semantics are built upon well-established tabular MDP methods that assume finite state and action spaces. Implementing our commitment semantics in infinite state and action spaces is a possible direction for future research. The MDP that the agent is in is drawn from the known prior distribution μ_0. If the agent is in MDP k, then by taking action $a_t \in \mathcal{A}$ in state $s_t \in \mathcal{S}_t$, it receives a reward $R_k(s_t, a_t)$ and the environment transits to $s_{t+1} \sim P_k(\cdot|s_t, a_t)$. It will be convenient if we let capitalized S_t be a random variable indicating the state at time step t whose specific realization is denoted s_t, and capitalized A_t be a random variable indicating the action being taken at time step t, whose specific realization is denoted a_t. Because the true MDP is partially observable, we consider history-dependent stochastic policies that map the history up to time step t,

$$h_t = \langle s_0, a_0, r_1, s_1, \ldots, s_{t-1}, a_{t-1}, r_t, s_t \rangle,$$

to a probability distribution over the next action. Specifically, we use $\pi(a|h)$ to denote the probability of choosing action a given history h when following policy π. During execution, the agent can use the information provided by the history so far to update the posterior distribution over the MDP it is actually facing. The posterior is a sufficient statistic that succinctly summarizes the agent's knowledge about the uncertain environment by interacting with it up to a certain point.

3.3 Prescriptive probabilistic commitment semantics

For the agents operating in the uncertain environment described in Sect. 3.2, Definition 1 formally gives our definition of a probabilistic commitment.

Definition 1 A *probabilistic commitment* is formally defined as a tuple

$$c = \langle \boldsymbol{\Phi}, T, \rho \rangle, \tag{2}$$

where $\boldsymbol{\Phi} \subseteq \mathcal{S}$ is the commitment state space, $T(\leq H)$ is the commitment time, and ρ is the commitment probability.

At a minimum, a probabilistic commitment in this form represents a prediction about how likely the state of the world will be an element of $\boldsymbol{\Phi}$ at time T, based on whatever policy the commitment provider is following. As summarized in Sect. 2, however, this kind of *predictive* commitment semantics can fail to match commonsense notions of what making a commitment means, since it does not in any way impede a provider from unilaterally changing its commitment whenever it chooses to alter its policy.

We therefore define a **prescriptive** semantics for a probabilistic commitment:

Definition 2 The *prescriptive probabilistic commitment semantics* for probabilistic commitment c requires that a commitment provider is constrained to follow a policy π, such that

$$\Pr_{k\sim\mu_0}(S_T \in \Phi | S_0 = s_0, k;\pi) \geq \rho. \tag{3}$$

By Eq. (3), our prescriptive probabilistic commitment semantics is clear: knowing that it is facing an MDP drawn from the prior distribution μ_0 over possible MDPs in the environment ($k \sim \mu_0$), the provider is constrained to follow a (in general history-dependent) policy, such that, starting at the initial state s_0, the probability of reaching a state in the commitment state space $S_T \in \Phi$ at the commitment time T is at least the commitment probability ρ. Unlike a predictive semantics, where the probability ρ of reaching a state in Φ at time T depends on the provider's choice of policy π, the prescriptive semantics turns the dependency around: the provider's choice of policy π depends on the committed probability ρ of reaching a state in Φ at time T.

Our commitment semantics generalizes the classical (logic-based) prescriptive semantics (Sect. 2) to settings where uncertainty precludes finding (even conditional) plans that provably reach states in the commitment state space Φ. Probabilistic commitments account for, and probabilistically quantify, the possibility that actions might have irreversible outcomes from which the commitment state space is unreachable. To satisfy our semantics, a provider should only agree to a commitment if it can formulate a policy with a sufficiently low probability ($< (1 - \rho)$) of such outcomes. If the provider then faithfully follows such a policy for the agreed-upon commitment, then by definition (Eq. 3) it has satisfied its commitment. Hence, a crucial consequence of our prescriptive probabilistic commitment semantics is that, now, **meeting a commitment is entirely under the agent's control** because satisfying a commitment only requires that the agent follow its policy in the states it finds itself in, rather than ensuring that specific states are guaranteed to be reached.[1]

Let Π_c be the set of all policies respecting the semantics of commitment c. We say that commitment c is feasible if and only if Π_c is not empty. Let

$$V^{\pi}_{\mu_0}(s_0) = E_{k\sim\mu_0}\left[\sum_{t=0}^{H-1} R_k(S_t, A_t) | S_0 = s_0;k,\pi\right]$$

be the expected cumulative reward starting at state s_0 under policy π if the true MDP is drawn from μ_0. We are interested in finding a policy that maximizes the expected cumulative reward while respecting the semantics of a given feasible probabilistic commitment, which is formally formulated as the following problem:

[1] For completeness, we should note that our semantics is also the probabalistic analogue of logic-based semantics for conditional commitments (Sect. 2). A conditional commitment asserts that a state in Φ will provably be reached in worlds where the specified conditions hold, but makes no promises when those conditions do not hold. As long as the agent's actions reach a state in Φ when the conditions hold, the commitment is satisfied. Analogously, a probabilistic commitment asserts that a state in Φ will be assuredly be reached whenever one out of the "good" subset of possible histories hold (where the probability of that occurring given the policy π is no less than ρ), but makes no promises otherwise. So, again analogously, as long as the agent takes actions prescribed by π, the commitment is met regardless of whether a state in Φ is reached in a specific episode.

$$\arg\max_{\pi \in \Pi_c} V^{\pi}_{\mu_0}(s_0). \tag{4}$$

Solving problem (4) involves two main challenges. First, it is non-trivial to characterize Π_c in a computationally-efficient manner that eases the policy optimization step. Second, under Bayesian uncertainty, finding the optimal policy (even without the constraint prescribed by the commitment semantics) requires planning with evolving posterior distributions. This imposes additional computational difficulty, since the number of posteriors grows exponentially with the time horizon. We propose methods in Sect. 4 that address these challenges.

4 Methods

In this section, we describe several methods for constructing policies with different trade-offs between solution quality and computational cost, while all the constructed policies are guaranteed to be in Π_c to respect the semantics of a given commitment c. In order to achieve high expected cumulative reward, the agent has to plan not only with fully observable states but also with the most recent knowledge about the true MDP it is in. Our first method, Commitment Constrained Full Lookahead (CCFL), finds the optimal policy in set Π_c by generating beforehand all possible posterior distributions over possible MDPs up to the finite time horizon. As a downside, since the number of posterior distributions generally grows exponentially as the time horizon grows, planning with all possible posterior distributions can make CCFL computationally infeasible. To this end, our Commitment Constrained Lookahead (CCL) method, generalizes CCFL by taking as input an integer parameter, L, as the number of time steps for posterior lookahead. Our Commitment Constrained No-Lookahead (CCNL) method can be treated as a special case of CCL, in which $L = 0$, and therefore actions are chosen only based on the initial conditions and ignoring posterior distributions. A small L often saves a lot of computational time compared to full lookahead, but by being more myopic decreases the expected cumulative reward. To partially mitigate this shortcoming of CCL (at the cost of a more modest increase in computation), we have created an iterative version of it called Commitment Constrained Iterative Lookahead (CCIL) that reapplies the CCL method in the midst of execution, where the posterior lookahead of successive applications of CCL reach closer to the time horizon.

4.1 Commitment constrained full lookahead

During execution, the agent can use the knowledge provided by the history so far to infer which MDP is more/less likely to be the true MDP it is facing. Formally, one can summarize current history h into a belief, $b := \langle s, \mu \rangle$, where s is the agent's current physical state, and μ is the posterior distribution over all possible MDPs given h. We use B_t to denote the random variable indicating the belief at time step t, and b_t to denote the belief given history h_t. The agent can find the optimal history-dependent policy by planning in the belief MDP defined as the tuple $\langle \mathcal{B}, \mathcal{A}, b_0, \tilde{P}, \tilde{R} \rangle$, where $\mathcal{B}$ is the set of all beliefs reachable from initial belief $b_0 = \langle s_0, \mu_0 \rangle$, which is finite because every possible true MDP k is finite and the time horizon is finite. $\tilde{P}$ and $\tilde{R}$ are belief transition and reward functions, respectively. Specifically, if we let $b|(a, r, s')$ be the belief after taking action a in belief state b, receiving reward r and transiting to state s', then the probability of transiting to any belief $b' \in \mathcal{B}$ after taking action a in belief state b can be expressed as

$$\max_{y} \quad \sum\nolimits_{b,a} y(b,a)\tilde{R}(b,a) \tag{6a}$$

$$\text{subject to} \quad \forall b,a \quad y(b,a) \geq 0; \tag{6b}$$

$$\forall b' \quad \sum\nolimits_{a'} y(b',a') = \sum\nolimits_{b,a} y(b,a)\tilde{P}(b'|b,a) + \delta(b',b_0); \tag{6c}$$

$$\sum\nolimits_{b:s\in\Phi} \sum\nolimits_{a} y(b,a) \geq \rho \tag{6d}$$

Fig. 3 CCFL program

$$\tilde{P}(b'|b,a) = \sum_{\{r,s':b|(a,r,s')=b'\}} \Pr(r,s'|b,a),$$

where $\Pr(r,s'|b,a)$ is the probability of receiving reward r and transiting to state s' after taking action a in belief b and can be expressed using $\{P_k,R_k\}_{k=1}^K$ as

$$\Pr(r,s'|b,a) = \Pr(r,s'|\langle s,\mu\rangle,a) = \sum_{k=1}^{K} \mu_k P_k(s'|s,a) 1_{\{r=R_k(s,a)\}}.$$

In words, given any belief $b' \in \mathcal{B}$, $\tilde{P}(b'|b,a)$ sums up probabilities over transitions (r,s') which update the belief to b'. Similarly, the belief reward function can be defined as

$$\tilde{R}(b,a) = \tilde{R}(\langle s,\mu\rangle,a) = \sum_{k=1}^{K} \mu_k R_k(s,a).$$

Our Commitment Constrained Full Lookahead (CCFL) method finds an optimal policy in Π_c among all belief-based policies, i.e., policies that choose actions as a function of the current belief, while respecting the commitment semantics. Note since a belief is a function of the history, then a belief-based policy also gives action probabilities as a function of the history. For MDP k, each policy π has a corresponding occupancy measure y_k^π for the expected number of times action a will be taken in belief-state b over the time horizon H:

$$y_k^\pi(b,a) = E\left[\sum_{t=0}^{H-1} 1_{\{B_t=b,A_t=a\}} | B_0 = b_0;k,\pi\right].$$

We will use shorthand notation y_k in place of y_k^π when policy π is clear from the context. If π is a belief-based policy, it can be recovered from its belief-action occupancy measure in any MDP k via

$$\pi(a|b) = \frac{y_k(b,a)}{\sum_{a'} y_k(b,a')}. \tag{5}$$

CCFL solves the mathematical program shown in Fig. 3, which introduces as decision variables the belief-action occupancy measure for all possible MDPs, and constructs the policy via Eq. (5) using the program's optimal solution. The CCFL program is a straightforward

adaptation of the linear program in Fig. 2 that solves an MDP. Constraints (6b) and (6c), which are the counterparts of constraints (1b) and (1c) in Fig. 2, guarantee that y is a valid occupancy measure with the initial belief being b_0 and the transition function being $\tilde{P}$. The expected cumulative reward is expressed using y in the objective function (6a), which is the counterpart of objective (1a). The commitment semantics of Eq. (3) imposes an additional constraint (6d), which ensures the resulting policy is in Π_c.

Because the belief is a sufficient statistic (i.e. it provides as much information for predicting the future as the history does), the CCFL program is feasible if the commitment is feasible, and the policy constructed by CCFL is optimal among all history-dependent policies respecting the commitment semantics, as formally stated in Theorem 1.

Theorem 1 *If commitment c is feasible, meaning $\Pi_c \neq \emptyset$, then the CCFL program in Fig. 3 is also feasible. Let y^* be an optimal solution to the CCFL program. The policy constructed via Eq. (5) using y^* is optimal with respect to the problem in Eq. (4).*

The proofs of theorems in this article are presented in the "Appendix".

4.2 Commitment constrained no-lookahead

Planning with all possible posterior distributions makes CCFL computationally infeasible, as confirmed by our empirical results in Sect. 6. To counter this, we now consider policies that ignore this posterior knowledge and only depend on the current state to choose actions. We refer to them as Markov policies and let Π_0 be the set of all Markov policies. If commitment c is feasible for Markov policies, i.e., $\Pi_c \cap \Pi_0 \neq \emptyset$, our Commitment Constrained No-Lookahead (CCNL) method will find an optimal Markov policy that maximizes expected cumulative reward respecting the commitment semantics, which is a solution to the following problem:

$$\arg\max_{\pi \in \Pi_c \cap \Pi_0} V^{\pi}_{\mu_0}(s_0). \tag{7}$$

Note that Π_0 is a subset of all history-dependent policies. When, as would generally be the case, Π_0 is a much smaller policy set, the computational cost of CCNL would be much less than that of CCFL, but the solution policy of CCNL is only an approximation of the optimal commitment semantics-respecting policy yielded by CCFL, as will be confirmed empirically in Sect. 6.

Similar to the belief-action occupancy measure, for MDP k, any policy π has a corresponding occupancy measure x^{π}_k of state-action pairs:

$$x^{\pi}_k(s,a) = E\left[\sum_{t=0}^{H-1} 1_{\{S_t=s, A_t=a\}} | S_0 = s_0; k, \pi\right].$$

We will use shorthand notation x_k in place of x^{π}_k when policy π is clear from the context. If π is a Markov policy, it can be recovered from its state-action occupancy measure in any MDP k via

$$\pi(a|s) = \frac{x_k(s,a)}{\sum_{a'} x_k(s,a')}. \tag{8}$$

$$\max_{x} \quad \sum\nolimits_k \mu_{0,k} \Big(\sum\nolimits_{s,a} x_k(s,a) R_k(s,a) \Big) \tag{9a}$$

$$\text{subject to} \quad \forall k,s,a \quad x_k(s,a) \geq 0; \tag{9b}$$

$$\forall k,s' \quad \sum\nolimits_{a'} x_k(s',a') = \sum\nolimits_{s,a} x_k(s,a) P_k(s'|s,a) + \delta(s',s_0); \tag{9c}$$

$$\forall k,k',s,a \quad \frac{x_k(s,a)}{\sum_{a'} x_k(s,a')} = \frac{x_{k'}(s,a)}{\sum_{a'} x_{k'}(s,a')}; \tag{9d}$$

$$\sum\nolimits_{s \in \Phi} \sum\nolimits_a \Big(\sum\nolimits_k \mu_{0,k} x_k(s,a) \Big) \geq \rho \tag{9e}$$

Fig. 4 CCNL program

CCNL constructs the policy by solving the mathematical program shown in Fig. 4. It introduces as decision variables the state-action occupancy measure for all possible MDPs. Constraints (9b) and (9c), as counterparts of constraints (1b) and (1c), guarantee that x_k is a valid occupancy measure with the initial state being s_0 and the transition function being P_k. The commitment semantics of Eq. (3) is explicitly expressed in constraint (9e). The expected cumulative reward is expressed using x in the objective function (9a), where $\mu_{0,k}$ is the probability that the true MDP is k according to μ_0. The corresponding Markov policy can be derived via Eq. (8). Unlike CCFL, the CCNL program is no longer a straightforward adaptation of the linear program in Fig. 2 because a challenging problem here is to ensure that these K sets of occupancy measures all derive the same Markov policy. To this end, we use constraint (9d) to enforce alignment across all K sets of occupancy measures. The constraints in Fig. 4 are feasible if and only if $\Pi_c \cap \Pi_0 \neq \emptyset$.

4.3 Commitment constrained lookahead

CCFL pre-plans for every possible revision to the agent's posterior knowledge about the true MDP it might be in, which guarantees optimality but possibly at a huge computational cost. At the other extreme, CCNL only considers Markov policies that ignore this evolving posterior knowledge. Here we consider the general case where the agent plans its first $L \in [0,H]$ actions as a function of the evolving belief, and thereafter plans actions based on the evolving state but with the belief (including both the state and the posterior distribution) the agent was in at time L. We refer to this parameter, L, as the belief-update lookahead boundary, which tells the planner how far beyond the current time to look ahead about states and posterior distributions. The resulting *L-updates* policy takes the form:

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

where b_t is the belief consistent with h_t, and b_L is the belief consistent with h_L when $t \geq L$. Note that a 0-update policy is the same as a Markov policy and an H-update policy is a full

$$\max_{x,y} \sum\nolimits_{b\in\mathcal{B}^{b_0}_{\leq L-1},a} y(b,a)\tilde{R}(b,a) + \sum\nolimits_{b_L\in\mathcal{B}^{b_0}_{L},k,s,a} x_{b_L,k}(s,a)R_k(s,a) \tag{12a}$$

subject to

$$\forall b \in \mathcal{B}^{b_0}_{\leq L}, a \quad y(b,a) \geq 0; \tag{12b}$$

$$\forall b' \in \mathcal{B}^{b_0}_{\leq L} \quad \sum\nolimits_{a'} y(b',a') = \sum\nolimits_{b,a} y(b,a)\tilde{P}(b'|b,a) + \delta(b',b_0); \tag{12c}$$

$$\forall b_L \in \mathcal{B}^{b_0}_{L} \quad y_{b_L} = \sum\nolimits_{a} y(b_L,a); \tag{12d}$$

$$\forall b_L \in \mathcal{B}^{b_0}_{L}, k, s, a \quad x_{b_L,k}(s,a) \geq 0; \tag{12e}$$

$$\forall b_L = \langle s_L, \mu_L\rangle \in \mathcal{B}^{b_0}_{L}, k, s' \quad \sum\nolimits_{a'} x_{b_L,k}(s',a') = \sum\nolimits_{s,a} x_{b_L,k}(s,a)P_k(s'|s,a) + \mu_{L,k}y_{b_L}\delta(s',s_L); \tag{12f}$$

$$\forall b_L \in \mathcal{B}^{b_0}_{L}, k, k', s, a \quad \frac{x_{b_L,k}(s,a)}{\sum_{a'} x_{b_L,k}(s,a')} = \frac{x_{b_L,k'}(s,a)}{\sum_{a'} x_{b_L,k'}(s,a')}; \tag{12g}$$

$$\sum\nolimits_{b_T\in\mathcal{B}^{b_0}_{T}:s_T\in\Phi,a} y(b,a) \geq \rho, \text{ if } T < L; \tag{12h}$$

$$\sum\nolimits_{b_L\in\mathcal{B}^{b_0}_{L},k,s\in\Phi,a} x_{b_L,k}(s,a) \geq \rho, \text{ if } T \geq L; \tag{12i}$$

Fig. 5 CCL program

width belief-based policy. Therefore, belief-update lookahead boundary L defines a continuum between CCNL and CCFL.

Given a specific value of L, let Π_L be the set of all L-updates policies. If commitment c is feasible for belief-update lookahead boundary L, i.e., $\Pi_c \cap \Pi_L \neq \emptyset$, our Commitment Constrained Lookahead (CCL) method will find an optimal L-updates policy that maximizes expected cumulative reward respecting the commitment semantics, which is a solution to the following problem:

$$\arg\max_{\pi\in\Pi_c\cap\Pi_L} V^{\pi}_{\mu_0}(s_0). \tag{10}$$

CCL constructs the policy by solving the mathematical program shown in Fig. 5, which is a novel and carefully-crafted combination of the techniques in CCFL and CCNL. The program introduces as decision variables y and x, where y is the belief-action occupancy measure (as defined for CCFL) for those beliefs reachable within the first L time steps of the plan, and x is the state-action occupancy measures (as defined for CCNL) for the remaining time steps to the horizon. We use $\mathcal{B}^b_l$ to denote the set of reachable beliefs after executing exactly l actions from belief b, and $\mathcal{B}^b_{\leq l} = \bigcup_{l=0}^{l'} \mathcal{B}^b_{l'}$ to denote the set of reachable beliefs from b by executing at most l actions starting from b. Because time is a state feature, $\mathcal{B}^b_l$ and $\mathcal{B}^b_{l'}$ are disjoint if $l \neq l'$. CCL generates beforehand all reachable beliefs from initial belief $b_{(t=)0}$ within L actions, $\mathcal{B}^{b_{(t=)0}}_{\leq L}$. The belief-action and state-action measures enable us

to express the expected cumulative reward very conveniently in the objective (12a) where the first term sums up the reward of the first L time steps, and the second term the remaining time steps to the horizon. The occupancy measures also enable us to express commitment semantics conveniently: if the lookahead does not reach the commitment time T, then the commitment semantics can be expressed in terms of the belief-action occupancy measure via constraint (12h); otherwise, the commitment constraint can be expressed in terms of those state-action occupancy measures via constraint (12i). Constraints (12b) and (12c) on y are the counterparts of (6b) and (6c) in the CCFL program of Fig. 3. Similarly, constraints (12e), (12f), and (12g) on x are the counterparts of (9b), (9c), and (9d) in the CCNL program of Fig. 4, which means the CCL program is considerably more sophisticated than the original linear program of Fig. 2. These constraints are feasible if and only if $\Pi_c \cap \Pi_L \neq \emptyset$. Any L-updates policy π_L that respects the commitment semantics can be derived from a feasible solution to the program in Fig. 5 via:

$$\pi_L(a|h_t) = \begin{cases} \pi_L(a|b_t) = \dfrac{y(b_t,a)}{\sum_{a'} y(b_t,a')} & t < L \\ \pi_L(a|s_t,b_L) = \dfrac{x_{b_L,k}(s_t,a)}{\sum_{a'} x_{b_L,k}(s_t,a')} & t \geq L \end{cases}. \tag{11}$$

Theorem 2 states that CCL using belief-update lookahead boundary L finds an optimal policy in $\Pi_c \cap \Pi_L$.

Theorem 2 *If $\Pi_c \cap \Pi_L \neq \emptyset$ holds for commitment c, then the program in Fig. 5 is feasible. Let x^*, y^* be its optimal solution, then the policy derived via Eq. (11) with x^*, y^* is the optimal policy in $\Pi_c \cap \Pi_L$.*

Intuitively, a belief-update lookahead boundary greater than zero enables the agent to plan actions not only based on the states it will visit, but also based on how its actions can provide information to improve its posteriors about what its true MDP is. Sacrifices in short-term reward may ultimately improve long-term performance. Theorem 3 says the expected cumulative reward of the policy derived by CCL using any $L > 0$ is lower bounded by that of the policy derived by CCNL. This is because, by definition, for any L and any Markov policy, there exists an L-updates policy that behaves exactly the same as the Markov policy, i.e. $\Pi_0 \subseteq \Pi_L$.

Theorem 3 *If $\Pi_c \cap \Pi_0 \neq \emptyset$ holds for commitment c, then for any integer $L \in [0, H]$ the CCL program in Fig. 5 is feasible, and we have*

$$V_{\mu_0}^{\pi_L^*}(s_0) \geq V_{\mu_0}^{\pi_0^*}(s_0)$$

where π_L^* and π_0^* are the policies derived by CCL using belief-update lookahead boundary L and zero, respectively.

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 MDPs vary solely in reward functions, but this monotonicity cannot be guaranteed in general, as stated in Theorems 4 and 5.

Theorem 4 *If MDPs vary in reward functions and not in transition dynamics, 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*

$$V_{\mu_0}^{\pi_L^*}(s_0) \leq V_{\mu_0}^{\pi_{L'}^*}(s_0)$$

where π_L^* *and* $\pi_{L'}^*$ *are the policies derived by CCL using boundaries* L *and* L', *respectively.*

Theorem 5 *There exists an environment, a commitment* c, *and boundaries* $0 < L < L' < H$ *satisfying* $\Pi_c \cap \Pi_L \neq \emptyset$ *and* $\Pi_c \cap \Pi_{L'} \neq \emptyset$, *such that*

$$V_{\mu_0}^{\pi_L^*}(s_0) > V_{\mu_0}^{\pi_{L'}^*}(s_0)$$

where π_L^* *and* $\pi_{L'}^*$ *are the policies derived by CCL using belief-updates boundaries* L *and* L', *respectively.*

These theoretical results provide some insights when choosing L. If the transition dynamics do not vary across MDPs, as suggested by Theorem 4, Π_L is monotonically increasing in L. One should use the largest affordable L because a larger L is likely to include more policies in Π_c and improve the value. A commitment that is infeasible for a smaller L could be feasible for a larger L. In general, though, the transition dynamics can vary across MDPs, and Π_L is not guaranteed to be monotonically increasing in L. One should use CCFL if it is affordable. CCFL considers all policies in Π_c if it is non-empty and therefore it yields optimal value. When CCFL is not affordable, then as suggested by Theorem 3 we can check the feasibility of a commitment with CCNL because a commitment feasible to CCNL (i.e. $\Pi_c \cap \Pi_0 \neq \emptyset$) is also feasible for any L. For our empirical results in Sect. 6, we experiment with several candidate values of L. Our experience suggests that L can best be chosen with problem-specific knowledge.

4.4 Commitment constrained iterative lookahead

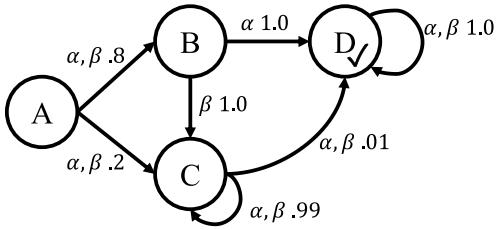
At each time step during execution, the agent observes the state transition that occurs and reward received to update its posterior μ about the true MDP it is in. One might think it would be a good idea for the agent to construct and follow an updated policy from its current state, substituting its updated belief state for the initial belief. However, the agent cannot shift from one policy to another without considering its commitment. Clearly, if the agent can find a plan that achieves the original commitment probability conditioned on the current belief, then shifting to such a plan will certainly respect the commitment semantics. Observation 1 says this re-planning is not always feasible.

Observation 1 *There exists an environment, a feasible commitment* c, *a policy* $\pi \in \Pi_c$, *and a history* h_t *induced by* π, *such that*

$$\forall \pi' \quad \Pr_{k \sim \mu_t}(S_T \in \Phi | S_t = s_t, k; \pi') < \rho,$$

where $\langle s_t, \mu_t \rangle$ *is the belief consistent with* h_t.

The example shown in Fig. 6 verifies Observation 1. Starting in state A, the agent can feasibly commit to reaching the absorbing state D at time step 2 with at least probability .8.



	R_1	R_2
α	0	1
β	1	0

Fig. 6 This example is adapted from Fig. 1. There are two possible reward functions R_1 and R_2 shown above with 50–50 prior. In both reward functions, the reward only depends on the action. There are two actions, α and β, and the transition dynamics is shown in the annotations of the edges. Starting in A, the agent commits to reaching the absorbing location D at time step two with at least probability .8. If the agent happens to be in C at time step one, there is no plan that reaches D from C with probability at least .8 (verifying Observation 1). Even though re-planning from C does not yield a plan that leads to D with probability 0.8, the new plan will nonetheless yield more reward because at time step one we will know which reward function applies and can therefore choose the more rewarding action in C

If the agent stochastically reached state C at time step 1, there is no plan that reaches state D from state C with probability at least .8, and this verifies Observation 1.

Our Commitment Constrained Iterative Lookahead (CCIL) method instead updates the commitment probability in a way that guarantees feasible re-planning, and iteratively applies CCL with that updated commitment probability during execution. The idea is that, when re-planning, respecting the commitment semantics does not require meeting the original probabilistic commitment, but instead to *fulfill the commitment probability that had originally been associated with the physical-state history traversed so far*.[2] Here we formally describe CCIL's first iterative application of CCL after having executed one or more actions. Suppose the agent now has belief $b_t = \langle s_t, \mu_t \rangle$ at time step $t \leq L$ after following policy π_L^* derived from the initial optimal solution to the CCL program with belief-update lookahead boundary L. Now the agent re-plans from s_t using its updated posterior b_t, with the commitment probability that its previous policy π_L^* ascribed to meeting the commitment if state s_t were reached:

$$\rho_t = \Pr_{k \sim \mu_t}(S_T \in \Phi | S_t = s_t, k; \pi_L^*). \tag{13}$$

Specifically, the agent constructs and follows a new L-updates policy, beginning from the current belief, by reusing the CCL program in Fig. 5 with the following modifications:

1. Start from current belief $b_t = \langle s_t, \mu_t \rangle$ instead of $b_0 = \langle s_0, \mu_0 \rangle$.
2. Let $L \leftarrow \min(L, H - t)$ to ensure that the lookahead from the current time step is bounded by the time horizon, i.e. $t + L \leq H$.
3. If the agent has not reached the commitment time, i.e. $t < T$, plan with the updated commitment probability by replacing ρ with ρ_t calculated as in Eq. (13) in constraint (12h) if $T < t + L$ or in constraint (12i) if $T \geq t + L$; otherwise, discard constraints (12h) and (12i) (e.g., let $\rho_t = 0$).

[2] We should point out that our earlier paper [43] that considered this Bayesian setting did not impose this constraint, instead insisting that whatever policy adopted from this point on, appended to the policy taken so far, would satisfy the commitment semantics if followed from the initial state. While that weaker constraint generally performed correctly, we identified corner cases where a dishonest commitment provider could exploit that constraint to increase its local reward. The constraint we provide here (also used in our more recent non-Bayesian paper [44]) closes this loophole.

Revisiting the example in Fig. 6, the initial policy could meet the commitment probability (0.8) by committing to take action α with probability 1 if B is reached at time 1, and otherwise the agent is unconstrained. After taking action α (or β) at time 0, then at time 1 the agent is either in B or C, and from the reward it just received knows the true reward function. Using CCIL, the agent re-plans. If it is in B, then since the original policy attributed probability 1 to meeting the commitment down this path, its new policy is constrained to take action α (whatever the true reward is), and afterwards take the better action. If it is in C, the updated commitment probability is zero (the original policy did not count at all on possibly meeting the commitment down this path), so the new policy can optimize reward without constraints.

In principle, the agent can iteratively apply the above procedure at any time during execution. We will evaluate empirically a version of CCIL that takes as input a pair of integers, (L, I), such that it iteratively uses L as the belief-update lookahead boundary to update the policy every $I \leq L$ steps. This procedure is outlined in Algorithm 1, and Theorem 6 proves that it respects our commitment semantics.

Theorem 6 *Let π_{IL} be the history-dependent policy defined as in Algorithm 1. We have $\pi_{IL} \in \Pi_c$.*

Algorithm 1: Commitment Constrained Iterative Lookahead (L, I)

Input: Environment $\mathcal{E} = \langle \mathcal{S}, \mathcal{A}, \{P_k, R_k\}_{k=1}^{K}, s_0, \mu_0 \rangle$,
commitment $c = \langle \Phi, T, \rho \rangle$,
integers $L \in [0, H], I \in (0, H]$ such that $\Pi_c \cap \Pi_L \neq \emptyset$ and $I \leq L$;
$b_0 \leftarrow \langle s_0, \mu_0 \rangle$;
$\pi_0 \leftarrow L$-updates policy derived by solving the program in Fig. 5;
$t \leftarrow 0$;
while $t < H$ **do**
  for $i = 1, 2, ..., I$ **do**
    Take action $a_t \sim \pi_t$ and observe reward-state transition $(s_t, a_t, r_{t+1}, s_{t+1})$;
    Update belief as $b_{t+1} = \langle s_{t+1}, \mu_{t+1} \rangle$;
    $\pi_{t+1} \leftarrow \pi_t$;
    $t \leftarrow t + 1$;
    if $t == H$ **then**
      Break the while loop;
    end
  end
  if $t < T$ **then**
    $\rho_t = \Pr_{k \sim \mu_t}(S_T \in \Phi | S_t = s_t, k; \pi_t)$;
  end
  else
    $\rho_t = 0$;
  end
  $\pi_t \leftarrow$ Policy derived by solving a modified version of the program in Fig. 5: let $L \leftarrow \min(L, H - t)$; replace every b_0 with b_t; replace ρ with ρ_t in constraint (12h) if $T < t + L$ or in constraint (12i) if $T \geq t + L$;
end

5 Dealing with the quadratic equality constraint

The CCFL program in Fig. 3 is a linear program straightforwardly adapted from the program in Fig. 2 and thus can be solved by standard linear programming algorithms. The CCL program in Fig. 5, however, is no longer a straightforward adaptation of Fig. 2 because it introduces a quadratic equality constraint (12g) to ensure that the action selection rules derived from occupancy measures in all possible MDPs are identical. Similarly, the CCNL program in Fig. 4 also introduces such a quadratic equality constraint (9d). These quadratic constraints makes the mathematical programs non-convex and hard to solve. In practice, many math-programming solvers are unable to handle programs with quadratic equality constraints (e.g., [8, 14]). Although some solvers can deal with such programs (e.g., [20, 23]), they often need to take as input a feasible solution as the starting point, but finding an initial feasible solution by itself might be difficult, and the final solutions are usually sensitive to starting points. Here we introduce two variant formulations of the CCL program in Fig. 5 that avoid quadratic equality constraints.

Deterministic CCL The policy derived from the program in Fig. 5 via Eq. (11) is in general stochastic. To enforce deterministic policies, Dolgov and Durfee [9, 10] introduced binary indicators in the linear programs for solving MDPs. Inspired by their work, we propose a novel formulation that avoids quadratic equality constraints by introducing binary indicators that force the action selection to be deterministic *after* belief-update lookahead boundary L. Specifically, we introduce indicators Δ as additional decision variables into the CCL program in Fig. 5 with the following constraints replacing the quadratic equality constraint (12g):

$$\begin{aligned}
&\forall b_L \in \mathcal{B}_L^{b_0}, s, a \quad \Delta_{b_L}(s,a) \in \{0,1\};\\
&\forall b_L \in \mathcal{B}_L^{b_0}, s \quad \sum_a \Delta_{b_L}(s,a) \leq 1;\\
&\forall b_L \in \mathcal{B}_L^{b_0}, k, s, a \quad x_{b_L,k}(s,a) \leq \Delta_{b_L}(s,a).
\end{aligned}$$

This reformulation yields a Mixed Integer Linear Program (MILP) which is well studied with many available solvers (e.g., [8, 14, 20, 23]). Any feasible solution with the above constraints replacing constraints (12g) of the program in Fig. 5 yields a policy with deterministic action selection at time steps after belief-update lookahead boundary L via Eq. (11), which can be alternatively expressed using the indicator variables:

$$\pi_L(a|h_t) = \begin{cases} \pi_L(a|b_t) = \dfrac{y(b_t,a)}{\sum_{a'} y(b_t,a')} & t < L \\ \pi_L(a|s_t,b_L) = 1_{\{\Delta_{b_L}(s_t,a)=1\}} & t \geq L \end{cases}. \tag{14}$$

Reward uncertainty only Quadratic equality constraint (12g) can be avoided when the transition dynamics does not vary across possible MDPs, i.e. $\forall k, k', P_k = P_{k'}$.[3] In this case, for the action selection at time step $t \in [H-L, H]$, without loss of optimality, the agent

[3] Our earlier work limited to reward uncertainty exploited this [43].

needs only to plan for the Bayes-optimal Markov policy w.r.t. the mean reward R_{μ_L} according to the belief it ended up in at time step L:

$$R_{\mu_L}(s,a) = \sum_k \mu_{L,k} R_k(s,a)$$

The resulting mathematical program is shown in Fig. 7. The main difference from the original CCL program in Fig. 5 is that it only introduces one occupancy measure x_{b_L} for each reachable belief b_L at time step L, instead of K sets of occupancy measures $\{x_{b_L,k}\}_{k=1}^K$ in the original CCL program. The derived policy can be expressed via:

$$\pi_L(a|h_t) = \begin{cases} \pi_L(a|b_t) = \dfrac{y(b_t,a)}{\sum_{a'} y(b_t,a')} & t < L \\ \pi_L(a|s_t,b_L) = \dfrac{x_{b_L}(s_t,a)}{\sum_{a'} x_{b_L}(s_t,a')} & t \geq L \end{cases}$$

6 Empirical results

As we summarized in Sect. 2, our work is the first to define a prescriptive semantics for probabilistic commitments, and develop algorithms that respect these semantics. Hence, in the empirical studies that follow, we predominantly focus on developing a deeper understanding of the strengths and limitations of different flavors of our algorithms. However, in an effort to illustrate empirically the difference between our approach and prior work, in our first study in the illustrative Windy L-Maze domain (Sect. 6.1), we compare to the closest related work we could identify: a non-prescriptive semantics for probabilistic commitments, and a prescriptive semantics for non-probabilistic commitments. We show how our prescriptive probabilistic commitment semantics allows agents to outperform either of these others because with it agents can balance selfish and unselfish behavior.

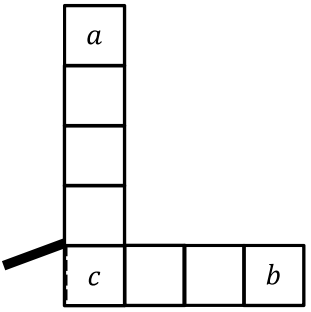
In Sect. 6.2, we use a small size Food-or-Fire domain to show how our CCL performs in an environment with both transition and reward uncertainty, and under various choices of belief-update lookahead boundary. In the subsequent two domains of RockSample (Sect. 6.3) and Change Detection (Sect. 6.4), the number of possible posterior distributions can grow so quickly with the time horizon that CCFL becomes computationally infeasible. In RockSample, we show how the iterative version of CCL, CCIL, is able to improve performance over CCL with modest additional computational cost. In Change Detection, we perform a detailed case study on the effects of the belief-update lookahead boundary and how it should be chosen with domain-specific knowledge, along with results reconfirming the improvement of CCIL over CCL.

6.1 Windy L-maze

The purpose of the experiments in this domain is to illustrate how our prescriptive probabilistic commitment semantics can improve multi-agent planning compared to alternative semantics. The domain consists of an L-maze occupied by a commitment provider and a recipient, as shown in Fig. 8.

$$
\begin{aligned}
\max_{x,y} \quad & \sum\nolimits_{b\in\mathcal{B}^{b_0}_{\leq L-1},a} y(b,a)\tilde{R}(b,a) + \sum\nolimits_{b_L\in\mathcal{B}^{b_0}_{L},s,a} x_{b_L}(s,a)R_{\mu_L}(s,a) \\
\text{subject to} \quad & \forall b \in \mathcal{B}^{b_0}_{\leq L}, a \quad y(b,a) \geq 0; \\
& \forall b' \in \mathcal{B}^{b_0}_{\leq L} \sum\nolimits_{a'} y(b',a') = \sum\nolimits_{b,a} y(b,a)\tilde{P}(b'|b,a) + \delta(b',b_0); \\
& \forall b_L \in \mathcal{B}^{b_0}_{L} \quad y_{b_L} = \sum\nolimits_{a} y(b_L,a); \\
& \forall b_L \in \mathcal{B}^{b_0}_{L}, s, a \quad x_{b_L}(s,a) \geq 0; \\
& \forall b_L = \langle s_L, \mu_L\rangle \in \mathcal{B}^{b_0}_{L}, s' \\
& \quad \sum\nolimits_{a'} x_{b_L}(s',a') = \sum\nolimits_{s,a} x_{b_L}(s,a)P(s'|s,a) + y_{b_L}\delta(s',s_L); \\
& \sum\nolimits_{b_L\in\mathcal{B}^{b_0}_{L},s\in\Phi,a} x_{b_L}(s,a) \geq \rho, \text{ if } L \leq T; \\
& \sum\nolimits_{b_T\in\mathcal{B}^{b_0}_{T}:s_T\in\Phi,a} y(b,a) \geq \rho, \text{ if } L > T;
\end{aligned}
$$

Fig. 7 CCL program in the reward uncertainty only case, i.e. $\forall k, k' \;\; P = P_k = P_{k'}$



d	0	1	2	3	4
R_1^p	0	0	0	0	0.1
R_2^p	-4	-3	-2	-1	0.1
R_3^p	4	3	2	1	0.1
R^r	3	0	0	0.1	0

Fig. 8 Windy L-maze. The provider starts in the cell labeled a and can only move in the vertical corridor, and the recipient starts in the cell labeled b and can only move in the horizontal corridor. It is admissible that both agents occupy the cell labeled c at the same time step. The table on the right specifies the reward functions, where d is the distance, measured by number of cells, between cell c and the provider/the recipient. For the provider, there are three possible reward functions $\{R_k^p\}_{k=1}^3$. The recipient's reward, R^r (bottom row), is known for certain

The provider starts in the cell labeled a and can only move in the vertical corridor, and the recipient starts in the cell labeled b and can only move in the horizontal corridor. It is admissible that both agents occupy the cell labeled c at the same time step. Let d^p, d^r be the distance, measured by number of cells, between cell c and the provider, the recipient, respectively. For the provider, there are three possible reward functions as functions of d^p, $\{R_k^p\}_{k=1}^3$, with a uniform prior:

$$\text{for } d^p = 4, R_1^p(d^p) = R_2^p(d^p) = R_3^p(d^p) = 0.1$$
$$\text{for } d^p < 4, R_1^p(d^p) = 0, R_2^p(d^p) = -R_3^p(d^p) = d^p - 4$$

The recipient's reward, R^r, is known as a function of d^r: $R^r(d^r) = 0.1$ if $d^r = 3$; $R^r(d^r) = 3$ if $d^r = 0$; $R^r(d^r) = 0$ for other values of d^r. The provider can move up, down, or stay in the current cell, and its moves succeed with probability one. The recipient can move left, right, or stay in the current cell. Initially, a door located in cell c is open with a strong wind blowing in such that the recipient's moves to the left only succeed with probability 0.1, and its other moves succeed with probability one. By occupying cell c, the provider can permanently close the door, in which case the wind stops and all the recipient's moves succeed with probability one. The two agents aim to maximize the joint expected reward up to the time horizon $H = 10$.

Because the recipient will get a significantly larger reward in cell c than in cell b, it is beneficial for the recipient if the provider could move to cell c to close the door. However, under reward functions R_1^p and R_2^p, traveling down the corridor to cell c will yield less reward for the provider than staying in the starting cell a. Therefore, effective coordination between the two agents is crucial to achieving high expected joint reward, where (as we shall see) the uncertain rewards of the provider make an "all-or-nothing" commitment suboptimal compared to a probabilistic commitment.

We compare the following three commitment semantics:

Non-Prescriptive Probabilistic Semantics In this case, a probabilistic commitment only represents a prediction of the provider's behavior [18, 41], rather than a prescription for how it will act. The provider computes and follows its history-dependent policy maximizing just its own local reward. It informs the recipient of the probability, $\underline{\rho}$, that the door will be closed at time step $T \geq 4$ under the provider's policy, and the recipient then computes and follows its own locally-optimal policy with respect to $\underline{\rho}$ by standard methods of solving MDPs. We refer to this semantics as *selfish* and *no-commitment* because the provider makes no effort to consider the preferences of the recipient when computing and executing its policy.

Prescriptive Non-Probabilistic Semantics This semantics is the logic-based semantics alluded to in work on detecting commitment abandonment [24], where a commitment provider will drop all else and single-mindedly pursue a commitment. In this case, the provider computes and follows its history-dependent policy that achieves the highest probability, $\overline{\rho}$, of closing the door at the earliest possible time step which is $T = 4$. The recipient uses $\overline{\rho}$ to compute and follow its optimal policy assuming maximum help from the provider. We refer to this semantics as *unselfish* and *full-commitment* because the provider prioritizes satisfying the preferences of the recipient over its own rewards.

Prescriptive Probabilistic Commitment This is the semantics we advocate in this paper. The provider makes a probabilistic commitment: it commits to closing the door at time step $T = 4$ with at least probability ρ. It uses the CCFL algorithm to compute and follow its locally-optimal policy that respects the commitment semantics. The recipient trusts this commitment, and computes and follows its optimal policy assuming the door will be closed at time step $T \geq 4$ with probability ρ.

The performance of each of the three different semantics (with a few choices of ρ for our prescriptive probabilistic semantics) is shown in Table 1. Notice that even when the

Table 1 Evaluation of non-prescriptive semantics, prescriptive non-probabilistic semantics, and prescriptive probabilistic commitment on the windy L-maze domain

Semantics	Provider	Recipient	Provider + recipient
Non-prescriptive probabilistic ($\underline{\rho} = 1/3$)	9.17	4.33	13.50
Prescriptive non-probabilistic ($\overline{\rho} = 1.0$)	4.90	10.61	15.51
Prescriptive probabilistic ($\rho = 0.6$)	9.06	6.84	15.90
Prescriptive probabilistic ($\rho = 0.7$)	8.62	7.79	16.41
Prescriptive probabilistic ($\rho = 0.8$)	7.38	8.73	15.61

The columns represent the cumulative rewards for the provider individually, the recipient individually, and both agents jointly

provider is acting entirely selfishly (the non-prescriptive probabilistic case), it predicts that it will nevertheless close the door with probability $\underline{\rho} = 1/3$. This is because its optimal policy is to move down the corridor one step, observe the reward signal to know exactly what the true reward function is, and then either go immediately back to a, or, with probability 1/3, it will learn that the reward function is R_3^p and continue on to c. Following the prescriptive non-probabilistic semantics, the unselfish provider will follow a policy guaranteed to close the door ($\overline{\rho} = 1.0$), because its moves succeed with certainty. With the prescriptive probabilistic commitment semantics, the agents can choose a probability of closing the door $\rho \in [0, 1]$ that balances selfishness and unselfishness in the provider to attain a higher joint reward. As ρ increases, the provider's value monotonically decreases and the recipient's value monotonically increases. As shown in Table 1, both $\rho = 0.6$ and $\rho = 0.8$ achieve higher joint reward than $\underline{\rho}$ and $\overline{\rho}$, and $\rho = 0.7$ is even better than $\rho = 0.6$ and $\rho = 0.8$.

These results confirm that our semantics for probabilistic commitments, coupled with algorithms for agent decision-making that respect these semantics, can lead to better joint performance than treating commitments either as inflexible logical constraints on the provider's plan (such that it must provably satisfy the commitment) or as non-binding predictions about the likelihood the agent's plan will happen to satisfy the commitment. Our semantics enable agents to strike a compromise between these extremes.

6.2 Food-or-Fire

The purpose of the experiment in this domain is twofold: 1) it is used to simply illustrate that CCL works well in an environment with both transition and reward uncertainty to construct policies respecting the semantics of a given probabilistic commitment, and 2) it is small enough that we can show the effect of the belief-update lookahead boundary by experimenting with all possible choices for the boundary from zero to the time horizon.

The environment is a simple two by three grid maze with $K = 3$ possible scenarios, as shown in Fig. 9, where solid black lines indicate impassable walls. The prior over the three scenarios is a uniform distribution. In the "empty" scenario, the agent can move freely in four directions within the maze, and no reward signal occurs. In the "food" scenario, there are two sections of impassable wall, and food associated with a reward of $+1$ exists in the mid-left cell between the walls. The "fire" scenario is the same as the second except that food is replaced with fire associated with a reward of -1. The agent, starting in the bottom

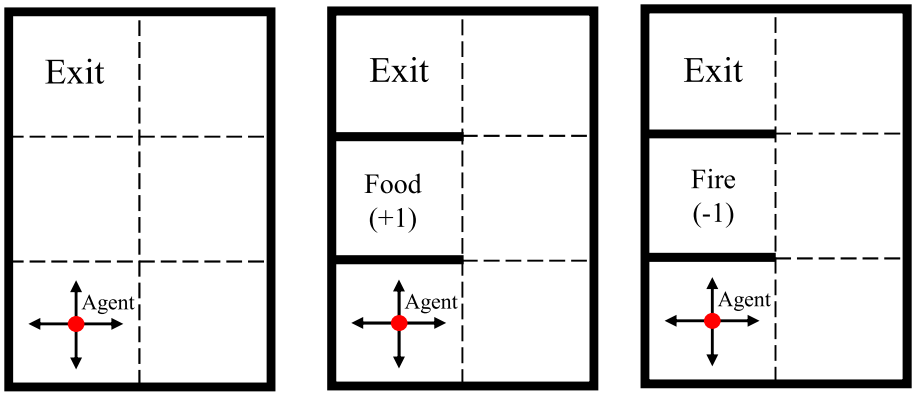


Fig. 9 Food-or-fire. Left: the "empty" scenario. Middle: the "food" scenario. Right: the "fire" scenario

left cell, commits to reach the top left cell (Exit) at the time horizon, i.e. $T = H$, with at least probability ρ. The agent can fully observe its current location but can only detect a wall by trying (and failing) to move between two adjacent cells.

Because the transition dynamics vary across the three scenarios, we only implemented deterministic CCL described in Sect. 5. Figure 10 plots the expected cumulative reward against all possible belief-update boundaries using deterministic CCL under various choices of T and ρ. According to Theorem 5, the monotonic performance in belief-update lookahead boundary L cannot be guaranteed, but it turns out the expected cumulative reward using deterministic CCL is monotonically non-decreasing with L for all choices of T and ρ we tried. Thus, anecdotally, it is not hard to find cases in which a larger L yields higher value, even though by Theorem 5 it is not guaranteed. Moreover, when L increases from two to three, we observe that the expected cumulative reward increases significantly for most choices of T and ρ. This is because a belief-update lookahead boundary L of three is just sufficient to identify which scenario the agent is actually facing by moving to the middle-left cell using three actions and reasoning about the observed reward signal of food, fire, or neither. Not surprisingly, with lower commitment probabilities, the agent is able to achieve higher expected reward. An interesting observation is that, compared with $\rho = 0.8$, we see the the expected cumulative reward is more like a step function at $L = 3$ for $\rho = 0.5$ and $\rho = 1.0$. When $\rho = 1.0$, the agent has to reach the Exit at time T in all three scenarios, so it suffices to determine the optimal behavior as soon as the agent figures out at time $L = 3$ which scenario it is facing. When $\rho = 0.5$, the agent would certainly reach the Exit in the "empty" scenario and the "fire" scenario. With the uniform prior, these two scenarios already contribute to $2/3 \geq \rho = 0.5$ probability of fulfilling the commitment, and therefore in the "food" scenario the agent would stay in the cell with food for the $+1$ reward and never exit. To achieve this behavior when $\rho = 0.5$, it suffices to use $L = 3$. For $\rho = 0.8$, it is more complicated in the sense that the agent also needs to reach the Exit with some positive probability in the second (food) scenario, and our results show that, with deterministic CCL, using L larger than 3 is able to improve the value.

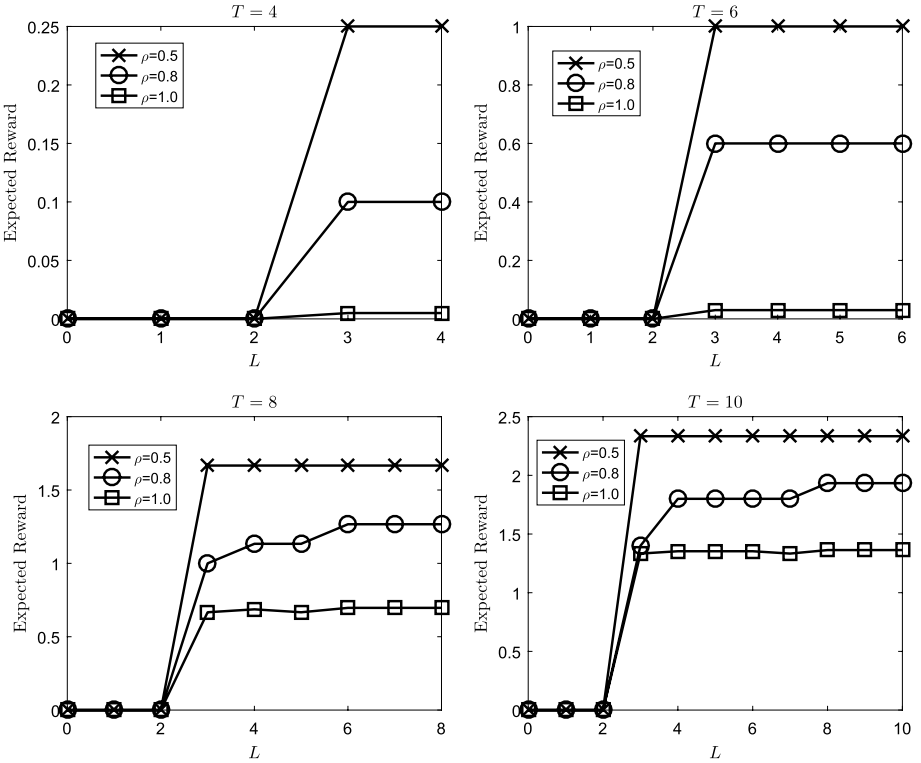


Fig. 10 Expected cumulative reward in Food-or-Fire domain as a function of the commitment and the belief-update lookahead boundary

6.3 RockSample

The size of the Food-or-Fire domain is small enough for us to afford computing belief-update boundaries up to the time horizon. In this RockSample domain and the following Change Detection domain, the number of posterior distributions grows so quickly as the time horizon grows that CCFL becomes computationally infeasible. Our results show that using the iterative version of CCL, CCIL, can improve the performance significantly with moderate additional computational cost.

RockSample [34] is a classic POMDP problem that models a rover exploring an unknown environment. In an instance of RockSample (n, s), the rover can move in an $n \times n$ grid containing s rocks. When n and s become large, a large belief-update lookahead boundary becomes computationally infeasible. The locations of the rocks are known. Only some of the rocks have scientific value and are of type *Good*; the others are of type *Bad*. The type of each rock is uniformly random. The task is to determine which rocks are valuable, approach and take samples of valuable rocks, and leave the map by moving off the right-hand edge of the map. Each time step, the rover can select from $s + 5$ actions: {*North*, *East*, *South*, *West*, *Sample*, $Check_1, \ldots, Check_s$}. Each $Check_i$ action directs the rover's sensor to rock i, returning a noisy observation from {*Good*, *Bad*}. The noise in the observations received by executing each $Check_i$ action is determined by the Manhattan distance

between the rover and the rock being checked: the probability of receiving a correct observation is 0.9, 0.7, and 0.5 when the the Manhattan distance is 0, 1, and at least 2, respectively. In an instance of RockSample (n, s), s rocks could have 2^s possible combinations of type assignments. We treat them as $K = 2^s$ possible MDPs that only differ in reward, and solve the program in Fig. 7 to construct CCL and CCIL policies. During execution, the observations from $Check_i$ actions are model-informative, suggesting which MDP is more likely.

In the original RockSample problem, the rover chooses actions to execute until it moves off the map and receives a positive reward. We adapted it to incorporate the probabilistic commitment: the rover does not receive any reward by moving off the map, but it has to move off the map by the time horizon, i.e. $T = H$, with at least the commitment probability ρ. We scale the reward to the range of $[-1, 1]$: the rover receives a reward of 1.0 for sampling a rock of type *Good*, a reward of $-$ 1.0 for sampling a rock of type *Bad*, and no reward occurs for re-sampling the same rock.

We evaluated CCL and CCIL on instances of RockSample (2, 2) and RockSample (4, 4) (Fig. 11). Table 2 contains the results of expected reward and run time in RockSample (2, 2) for commitment time $T = 10$ and commitment probability $\rho = 1.0$ with various choices of L and I. The run time for CCIL is the sum of the CPU times for each iteration. Note that because 1) the rover can get pretty accurate observations since it is always close to the rocks, 2) the types of rocks are uniformly random, and 3) time horizon 10 is large enough, the optimal behavior can collect in expectation one good rock, yielding an expected cumulative reward close to 1.0. For CCL, the results in Table 2 indicate that a larger belief-update lookahead boundary indeed improves the expected reward, but the computational time also increases dramatically. We can see that CCIL can achieve comparable expected reward with much less computational time than CCL. Although CCIL ($L = 3, I = 1$), CCIL ($L = 4, I = 4$), and CCL ($L = 8$) all achieve near-optimal expected reward, CCIL ($L = 3, I = 1$) and CCIL ($L = 4, I = 4$) use much less computational time than CCL ($L = 8$).

Table 3 contains the results in RockSample (4, 4) for commitment time $T = 15$ and probability $\rho = 1.0$. With $T = 15$, the time is just enough for the rover to correctly detect 3 rocks, sample the good rocks, and move off the map. Since a rock is good with probability .5, the expected cumulative reward of the optimal behavior is close to 1.5. For RockSample (4, 4), we can see that CCL can only scale to relatively small belief-update boundaries. The computational time grows dramatically, and we run out of memory when $L = 5$. CCL achieves an expected cumulative reward of 0.9 for $L = 4$, which means that a larger L is needed to find the near-optimal behavior. CCIL performs much better than CCL because it iteratively re-plans during the execution. The performance of CCIL ($L = 1, I = 1$) is between that of CCL ($L = 3$) and CCL ($L = 4$). CCIL ($L = 2, I = 2$), CCIL ($L = 2, I = 1$), and CCIL ($L = 3, I = 3$) all achieve behavior with expected cumulative reward close to 1.3, which cannot be achieved by CCL using a moderate amount of computational time. These three choices of (L, I) achieve comparable expected reward (no statistically significant difference), with CCIL ($L = 2, I = 2$) being the fastest because its iterative lookahead is less frequent than CCIL ($L = 2, I = 1$) and shallower than CCIL ($L = 3, I = 3$).

6.4 Change detection

In Change Detection, we perform a detailed case study on the effects of the belief-update lookahead boundary, where time horizon H is short enough so that we can experiment with

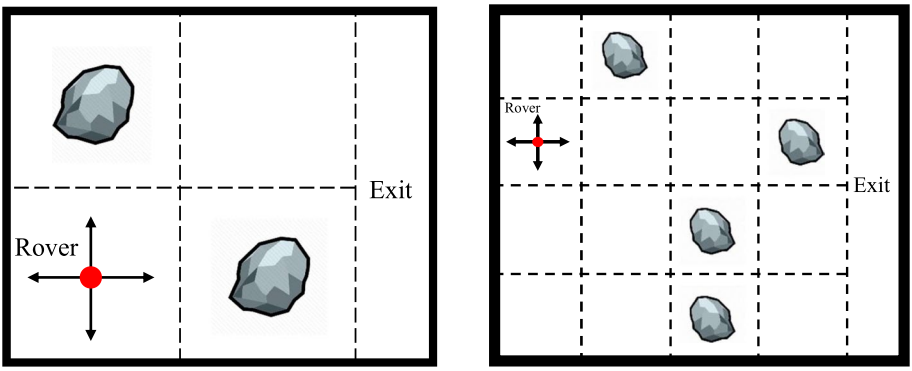


Fig. 11 RockSample instances: **a** RockSample (2,2), **b** RockSample (4,4)

Table 2 Results on RockSample (2,2), $|\mathcal{S}| = 177, |\mathcal{A}| = 7, |\mathcal{O}| = 4$ with $T = 10, \rho = 1.0$

L	I	Expected reward	Time (s)
0	n.a.	0.00	0.30
1	n.a.	0.20	0.54
2	n.a.	0.40	1.07
3	n.a.	0.60	3.05
4	n.a.	0.64	7.53
6	n.a.	0.82	45
8	n.a.	0.90	710
10	n.a.	n.a.	> 1000
1	1	0.53 ± 0.02	4.83 ± 0.28
3	1	1.01 ± 0.02	33.89 ± 1.67
3	3	0.81 ± 0.02	7.73 ± 0.13
4	1	0.97 ± 0.02	133.11 ± 10.67
4	4	0.92 ± 0.02	17.55 ± 0.30

1000 s run time limit

every $L \leq H$ for CCL. We also experiment with a larger H for which CCFL is computationally infeasible, to develop further intuitions about balancing lookahead with iteration to achieve good performance with reasonable computation.

Change Detection is a classic constrained POMDP problem [30]. The agent can partially observe the environment, and at some point the environment will transit into a state where the alarm should be sounded by the agent. The agent aims to minimize the delay in alerting (sounding the alarm) after the transition, and the probability of a false alarm should be lower than a given threshold which is referred to as the false alarm (F.A.) tolerance. Formally, the state space and action space are $\mathcal{S}$={*PreChange*, *PostChange*, *PostAlarm*, *FalseAlarm*}, $\mathcal{A}$={*NoAlarm*, *Alarm*}, respectively. The environment starts in *PreChange*, and transits to *PostChange* at a random time step if the agent has not performed action *Alarm*. Specifically, the problem has a geometric change time parameter η, such that at every time step, if the state is still *PreChange*, it will transit to *PostChange* with probability η. Once the agent performs action *Alarm*, the state transits to *PostAlarm* from *PostChange* with a positive reward, or to *FalseAlarm* from *PreChange* with no reward. The commitment is to

Table 3 Results on RockSample (4, 4), $|\mathcal{S}| = 4097, |\mathcal{A}| = 9, |\mathcal{O}| = 8$, with $T = 15, \rho = 1.0$

L	I	Expected reward	Time (s)
0	n.a.	0.00	4.33
1	n.a.	0.30	5.11
2	n.a.	0.30	8.71
3	n.a.	0.60	23.36
4	n.a.	0.90	113
5	n.a.	Out of memory	n.a.
1	1	0.74 ± 0.02	83.06 ± 0.55
2	1	1.32 ± 0.02	482.30 ± 31.53
2	2	1.31 ± 0.02	132.17 ± 3.73
3	1	n.a.	> 1000
3	3	1.34 ± 0.02	634.27 ± 67.37

1000 s run time limit

not reach *FalseAlarm* with at least a given probability. To encourage early detection, the agent receives a reward of + 1.0 if it executes action *Alarm* immediately after transiting to *PostChange*, with the reward discounted each subsequent time step. The states are not fully observable. Instead, the agent makes an observation o every time step from the observation space $\mathcal{O}$, suggesting if the environment has changed or not. The probability of making a specific observation is determined by probability mass functions $f_0, f_1 : \mathcal{O} \mapsto [0, 1]$ when the environment is in *PreChange*, and *PostChange*, respectively. In our experiments, the agent can make an observation every time step from a set of size $|\mathcal{O}| = 3$. The reward discount factor is set to $\gamma = 0.8$. The *PreChange* and *PostChange* observation distributions are

$$f_0(o_1) = 0.6, f_0(o_2) = 0.3, f_0(o_3) = 0.1,$$
$$f_1(o_1) = 0.2, f_1(o_2) = 0.4, f_1(o_3) = 0.4.$$

Parameter η provides the agent with the prior distribution of the change time. After making observations, the agent can use Bayes' rule to calculate the posterior distributions.

We consider the finite horizon decision problem, with the commitment time $T = H$ being equal to the time horizon, and define the state of the Change Detection problem as $s = \langle t, Alarmed \rangle$ where *Alarmed* is a Boolean that takes the value of true when the agent executed action *Alarmed* in any time step before t, or false otherwise. The current time step t and Boolean *Alarmed* are both fully observable to the agent. We define belief as $b = \langle s, \mu \rangle$, where state s is augmented by probability mass function μ that gives the probability of all possible change times up to the horizon.

Figure 12 contains the results when experimenting with CCL on a Change Detection instance with horizon $H = T = 10$, where CCFL is computationally feasible. We have experimented with two choices of the geometric change time parameter, $\eta = 0.1, 0.2$, and four choices of the false alarm (F.A.) tolerance. When F.A. tolerance is 0.0, the agent is forbidden to execute *Alarm* actions if there is any possiblity of false alarm, and therefore the expected cumulative reward is 0 for any choice of the belief-update lookahead boundary L. Otherwise, the expected cumulative reward is monotonically increasing with L. Moreover, choosing a large L is most helpful when the geometric change time parameter η is small (Fig. 12left). For $\eta = 0.1$ (Fig. 12left), the expected reward rises anywhere from about 3-fold (for tolerance=0.2) to 7-fold (for tolerance=0.05), while for $\eta = 0.2$ (Fig. 12right) it is anywhere from about 1.5-fold (for tolerance = 0.2) to 3.5-fold (for tolerance = 0.05).

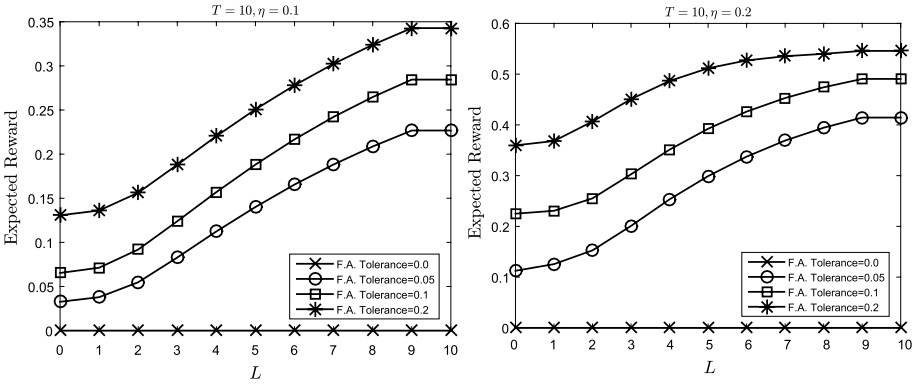


Fig. 12 Results of CCL on change detection with $T = 10, \gamma = 0.8$

So for the same tolerance, lookahead makes twice the impact when $\eta = 0.1$ than $\eta = 0.2$. Small η suggests that the change is more likely to happen later, and therefore a large L is more likely to envision it. For both choices of η, as lookahead L increases, the relative increase in expected reward is smaller when F.A. tolerance is larger. This is because larger tolerance inherently gets more reward regardless of lookahead, and hence there is less reward for lookahead to recoup. These results suggest that, more generally, the value of L should be chosen based at least upon: (1) how far into the future the most meaningful changes to the belief state will occur (as captured by η in this case), (2) how sensitive the agent's reward is to making a more informed decision (as captured by F.A. tolerance in this case), and (3) how dramatically computation costs rise with farther lookahead (where in this case the branching factor of 2 (change or no change) is fairly low).

We have also experimented with a larger horizon, $H = T = 50$, where CCFL is not computationally affordable. The geometric change time parameter is $\eta = 0.04$. As we just saw, a low value like this makes the change more likely to happen later and thus emphasizes farther lookahead. The F.A. tolerance is set to 0.2. Table 4 contains the results of expected reward and run time for CCL and CCIL with various choices of L, and of I when applicable. The run time of CCL grows dramatically with L. The expected reward, though, grows relatively slowly, because these lookaheads are still very short for such a small η that requires large lookahead. This can be inferred from Fig. 12(left), where $\eta = 1.0$ is larger and we still see a steep increase in reward at $L = H/2$. Nevertheless, there is still a 3-fold increase in reward when we increase L for CCL until the computation budget is reached. For CCIL, we experiment with $L = 2, 4, 6$ and $I = 1, L/2, L$. Unsurprisingly, with more frequent iterative lookahead (smaller I), both the expected reward and the run time increase. CCIL ($L = 4, I = 1$) achieves reward that is higher than any CCL within the computation budget. Both CCIL ($L = 6, I = 1$) and CCIL ($L = 6, I = 3$) double the reward of CCL ($L = 10$), the largest L within the computation budget, yet use much less computation. These results verify again the effectiveness of the iterative lookahead strategy in CCIL. Recall that, in RockSample, setting $I = L$ achieves significantly larger reward than CCL with the same L. However, in Change Detection, $I = L$ achieves no higher reward than CCL for the values of L we consider. We conjecture that this is because the belief changes frequently in Change Detection (every time step) and perhaps in a way that is critical for the agent's future decisions, making it necessary to perform frequent iterative lookahead, while it might take several steps in RockSample to experience a change (after taking the $Check_i$ action). From the results of $L = 4, 6$ and $I = 1, L/2$, we observe that

Table 4 Results on change detection with F.A. tolerance of 0.2, $T = 50, \eta = 0.04, \gamma = 0.8$

L	I	Expected reward	Time (s)
1	n.a.	0.05	0.02
2	n.a.	0.06	0.05
3	n.a.	0.07	0.16
4	n.a.	0.09	0.46
6	n.a.	0.11	4.23
9	n.a.	0.15	125
10	n.a.	0.16	761
11	n.a.	n.a.	> 1000
2	1	0.06 ± 0.02	1.62 ± 0.08
2	2	0.04 ± 0.02	0.99 ± 0.04
4	1	0.28 ± 0.04	16.33 ± 0.98
4	2	0.17 ± 0.04	9.85 ± 0.78
4	4	0.09 ± 0.03	4.04 ± 0.30
6	1	0.32 ± 0.03	117.11 ± 7.84
6	3	0.31 ± 0.04	33.01 ± 2.56
6	6	0.13 ± 0.04	28.41 ± 1.98

1000 s run time limit

with larger L, the agent can use larger I without sacrificing too much reward. Overall, CCIL ($L = 4, I = 1$) and CCIL ($L = 6, I = 3$) achieve the best compromise for a wide range of tradeoffs between solution quality and computational cost.

7 Conclusion

In this article, we argue in favor of an operational semantics we defined for a commitment provider that is operating under model uncertainty. Our semantics is based on what a commitment provider can control—its own actions. Specifically, we considered a decision-theoretic setting where the agent is making sequential decisions in one out of several MDPs with a known prior. Fulfilling a commitment corresponds to pursuing a course of action, beginning at the time the commitment was made and over the known prior, that has sufficient likelihood of achieving the intended state at a certain time prescribed by the commitment. In this semantics, the agent fulfills its commitment by following a commitment-constrained policy even if, due to bad luck, the desired outcome was not realized. Based on this semantics, we developed Commitment Constrained Lookahead (CCL), a novel algorithm parameterized by the belief-update lookahead boundary, that constructs semantics-respecting policies offline for the provider. We empirically compared our new semantics, operationalized in CCL, with prior logical and predictive semantics concepts, to illustrate where and why our semantics is superior. We also analytically and empirically investigated the impact of the belief-update lookahead boundary that makes an explicit tradeoff between the computation cost and performance of the computed policy. We have further extended CCL to Commitment Constrained Iterative Lookahead (CCIL) that iteratively adjusts the policy online according to the evolving posterior distribution about the true environment, while still respecting the commitment semantics. Our empirical results show that CCIL can achieve the same performance as CCL with much less computation overhead.

There are a number of interesting directions for future work. As we have mentioned, developing algorithms that incorporate our commitment semantics in infinite state and action spaces is a possible direction for future work. Moreover, our emphasis in this article has been on how the provider should constrain its behaviors for trustworthy commitment achievement. In open systems where trust needs to be earned, interesting questions arise as to how easy it would be to verify if an agent has acted in good faith on the commitment, where the outcomes of the decisions are observable but the decisions themselves are not. With the provider's trustworthy achievement, to make the commitment useful we also need to answer the question of how the recipient should properly model the commitment and plan accordingly. Finally, with semantics and mechanisms for representing, pursuing, and modeling a given commitment, we are prepared to answer the question of what commitment cooperating agents should agree to make.

Acknowledgements Funding was provided by Air Force Office of Scientific Research (Grant No. FA9550-15-1-0039). We thank the anonymous reviewers for their thoughtful comments.

Appendix

Here we present all the technical proofs of the theorems in this article.

Proof of Theorem 1 Note that the belief is a sufficient statistic: given history h_t at time step t and the corresponding belief b_t consistent with h_t, one does not need any other information in h_t besides b_t to predict the future state transitions and reward after time step t. Therefore, solving problem (4) is equivalent to solving a constrained MDP, where the MDP is the belief MDP defined as the tuple $\langle \mathcal{B}, \mathcal{A}, b_0, \tilde{P}, \tilde{R} \rangle$ with finite state space of beliefs, and the constraint comes from the semantics of commitment c. Our CCFL method can be viewed as a standard linear programming approach to solving a finite state constrained MDP.

□

Proof of Theorem 2 It is sufficient to show (1) any policy in $\Pi_c \cap \Pi_L$ can be derived from a feasible solution to the program in Fig. 5, and (2) any feasible solution to the program derives a policy in $\Pi_c \cap \Pi_L$.

To show (1), for any policy $\pi \in \Pi_c \cap \Pi_L$, we are going to define vectors m^π and n^π such that with m^π treated as x and n^π treated as y, m^π and n^π satisfy the constraints of the program in Fig. 5, and the L-updates policy π can be derived via Eq. (11). Specifically, given any policy $\pi \in \Pi_c \cap \Pi_L$, let n^π be its belief-action occupancy measure for beliefs in $\mathcal{B}^{b_0}_{\leq L}$, and m^π be its state-action occupancy measure for states from time step L on:

$$\forall b \in \mathcal{B}^{b_0}_{\leq L}, a \quad n^\pi(b,a) = \Pr(B_t = b, A_t = a | B_0 = b_0; \pi)$$

where t is the time of belief b, and

$$\forall s, a \quad m^\pi_{b_L,k}(s,a) = \begin{cases} \Pr(S_t = s, A_t = a, B_L = b_L, k | B_0 = b_0; \pi) & t \geq L \\ 0 & t < L \end{cases}$$

where t is the time of state s. Then, with m^π treated as x and n^π treated as y, m^π and n^π satisfy the constraints of the program in Fig. 5, and the L-updates policy π can be derived via Eq. (11).

To show (2), given a feasible solution x, y to the program, let policy π be the derived policy via (11). Then π is in Π_L by definition. Further we have $m^{\pi}_{b_L,k}(s,a) = x_{b_L,k}(s,a), n^{\pi}(b,a) = y(b,a)$, where m^{π} and n^{π} are defined as above. Therefore π is also in Π_c because x satisfies commitment constraints (12i), (12h). □

Proof of Theorem 3 By Theorem 2, CCL with boundary L finds the optimal policy in $\Pi_c \cap \Pi_L$. Therefore, it is sufficient to show

$$\forall L > 0, \Pi_0 \subseteq \Pi_L.$$

This holds because given any Markov policy $\pi_0 \in \Pi_0$ we can define an L-updates policy $\pi_L \in \Pi_L$ that is equivalent to π_0:

$$\pi_L(a|h_t) = \begin{cases} \pi_L(a|b_t) = \pi_0(a|s_t) & t < L \\ \pi_L(a|s_t, b_L) = \pi_0(a|s_t) & t \geq L \end{cases}.$$

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

Proof of Theorem 4 It is sufficient to show that the statement holds when $L' = L + 1$. We next show that when $P_k = P_{k'}\ \forall k, k'$, given any policy $\pi_L \in \Pi_L$, there exists an $(L+1)$ -updates policy, π_{L+1}, that mimics π_L , and therefore $V^{\pi_L}_{\mu_0}(s_0) \leq V^{\pi_{L+1}}_{\mu_0}(s_0)$.

For the first L actions, an $(L+1)$-updates policy can map the current belief to a distribution of the next actions identical to π_L, and the action that is going to be taken at time step L by π_L can also be recovered by an $(L+1)$-updates policy, which gives

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

Under any L-updates policy π_L, and conditioned on being in belief b_{L+1} at time step $L+1$, the agent thereafter selects actions according to $\pi_L(\cdot|s_t, b_L)$ with probability that the agent was in belief b_L at time step L: $\Pr(b_L|b_{L+1};\pi_L)$. If the transition dynamics does not vary across MDPs in the environment, it is well known [26] that a Markov policy $\pi_{b_{L+1}}(\cdot|s_t), t \geq L+1$ is sufficient to recover the state occupancy measure of π_L starting at belief b_{L+1}. Then π_{L+1} can also recover π_L for $t \geq L+1$ by demonstrating that $\pi_{b_{L+1}}$ satisfies

$$\pi_{L+1}(a|h_t) = \pi_{L+1}(a|s_t, b_{L+1}) = \pi_{b_{L+1}}(a|s_t) \qquad \text{for } t \geq L+1.$$

This concludes the proof. □

Proof of Theorem 5 In the proof of Theorem 4, we have shown that for any L-updates policy π_L there exists an $(L+1)$-update policy that is able to mimic π_L up to time step $L+1$. Provided that $P_k = P_{k'}\ \forall k, k'$, one can find a Markov policy that mimics π_L starting at any belief at time step $L+1$. When $P_k = P_{k'}\ \forall k, k'$ does not hold, however, this Markov policy in general does not exist, and therefore no $(L+1)$-update policy is able to mimic π_L. Inspired by this, we next give an example as a formal constructive proof.

Consider the example shown in Fig. 13. The environment has 10 locations $\{0, 1, \ldots, 9\}$, action space $\{up, down\}$, time horizon $T = 4$, and $K = 2$ possible MDPs. The agent starts in location 0 at time step $t = 0$ with a prior probability of 0.8 for MDP $k = 1$ and a prior probability of 0.2 for MDP $k = 2$. In MDP $k = 1$, no matter which action the agent takes, it transits to location 1 or 2 uniformly at random at time step $t = 1$, and then to location 3 with

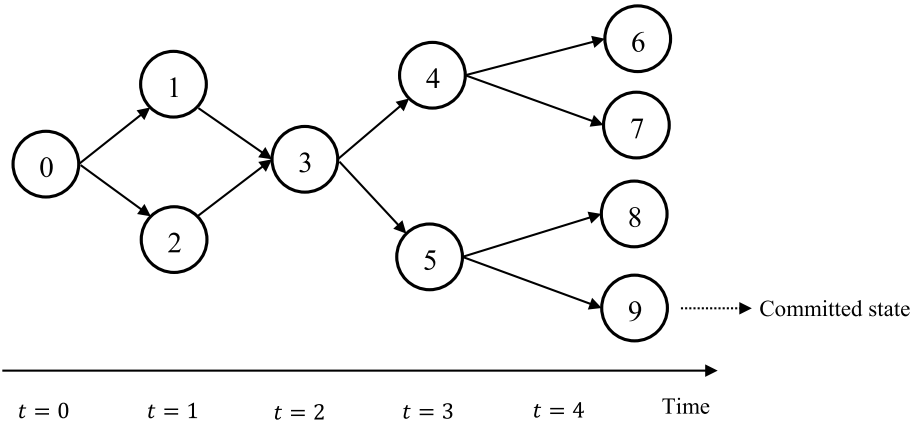


Fig. 13 Example as a proof of Theorem 5

probability one at time step $t = 2$. Starting from location 3, on taking action *up* (*down*) the agent transits to the upper (lower) location to the right. The transition dynamics of MDP $k = 2$ is the same as MDP $k = 1$ until the agent reaches location 3, and thereafter the transition is flipped: starting from location 3, on taking action *up* (*down*) the agent transits to the lower (upper) location to the right. In both MDPs, the agent will receive large negative reward ($-\infty$) in location 7 and 8. In MDP $k = 1$, the agent will receive $+1$ reward if it reaches location 6. There is no reward elsewhere. The agent commits to reaching location 9 with probability 0.5. Consider the following $(L =)1$-updates policy: if the agent was in location 1 at time step $t = 1$, always choose action *up*; if the agent was in location 2 at time step $t = 1$, always choose action *down*. Under this $(L =)1$-updates policy the probability of reaching the commitment location 9 is 0.5 and the expected reward is $0.8 \times 0.5 \times 1 = 0.4$. Now consider $(L =)2$-updates policies. Because the agent is in location 3 with probability one at time step $t = 2$. An $(L =)2$-updates policy amounts to a Markov policy for time steps $t \geq 2$. Further the agent should minimize the probability of reaching location 7 and 8 that yields large negative reward. One can verify that the only Markov policy for time steps $t \geq 2$ that avoids reaching location 7 and 8 while respecting the commitment semantics is to always choose action *down*, whose expected reward is 0, smaller than that of the $(L =)1$ -updates policy. □

Proof of Theorem 6 We need to show π_{IL} satisfies Eq. (3), i.e.,

$$\Pr_{k\sim\mu_0}(S_T \in \Phi | S_0 = s_0, k; \pi_{IL}) \geq \rho.$$

Let π_L be the CCL L-updates policy derived from the program in Fig. 5. The above inequality holds because:

$$
\begin{aligned}
&\Pr_{k\sim\mu_0}(S_T \in \Phi | S_0 = s_0, k; \pi_{IL}) \\
&= \sum_{b_I \in \mathcal{B}_I^{b_0}} \Pr_{k\sim\mu_0}(B_I = b_I | S_0 = s_0, k; \pi_{IL}) \Pr(S_T \in \Phi | B_I = b_I; \pi_{IL}) \\
&\quad \text{(law of total probability)} \\
&= \sum_{b_I \in \mathcal{B}_I^{b_0}} \Pr_{k\sim\mu_0}(B_I = b_I | S_0 = s_0, k; \pi_{L}) \Pr(S_T \in \Phi | B_I = b_I; \pi_{IL}) \\
&\quad (\pi_L \text{ and } \pi_{IL} \text{ are identical in the first } I \text{ steps}) \\
&\geq \sum_{b_I \in \mathcal{B}_I^{b_0}} \Pr_{k\sim\mu_0}(B_I = b_I | S_0 = s_0, k; \pi_{L}) \Pr(S_T \in \Phi | B_I = b_I; \pi_{L}) \\
&= \Pr_{k\sim\mu_0}(S_T \in \Phi | S_0 = s_0, k; \pi_L) \qquad \text{(law of total probability)} \\
&\geq \rho \qquad (\pi_L \in \Pi_c)
\end{aligned}
$$

The first inequality holds because CCIL iteratively applies L-step lookahead with the commitment probability achieved by the policy of the previous iteration. This concludes the proof. □

References

1. Agotnes, T., Goranko, V., & Jamroga, W. (2007). *Strategic commitment and release in logics for multi-agent systems (extended abstract)*. Technical Report IfI-08-01, Clausthal University.
2. Al-Saqqar, F., Bentahar, J., Sultan, K., & El-Menshawy, M. (2014). On the interaction between knowledge and social commitments in multi-agent systems. *Applied Intelligence*, *41*(1), 235–259.
3. Altman, E. (1999). *Constrained Markov decision processes* (Vol. 7). Boca Raton: CRC Press.
4. Bannazadeh, H., & 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.
5. Castelfranchi, C. (1995). Commitments: From individual intentions to groups and organizations. In *Proceedings of the international conference on multiagent systems* (pp. 41–48).
6. Chesani, F., Mello, P., Montali, M., & Torroni, P. (2013). Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems*, *27*(1), 85–130.
7. Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, *42*(2–3), 213–261.
8. CPLEX: IBM ILOG CPLEX 12.1. https://www.ibm.com/analytics/cplex-optimizer.
9. Dolgov, D., & Durfee, E. (2005). Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In *International joint conference on artificial intelligence* (Vol. 19, pp. 1326–1331).
10. Dolgov, D. A., & Durfee, E. H. (2004). Optimal resource allocation and policy formulation in loosely-coupled Markov decision processes. In *Proceedings of the fourteenth international conference on automated planning and scheduling* (pp. 315–324).
11. Durfee, E. H., & Singh, S. (2016). On the trustworthy fulfillment of commitments. In *Autonomous agents and multiagent systems: AAMAS 2016 workshops best papers.* (pp. 1–13). Springer lecture notes in artificial intelligence (2016). Also in Notes of the AAMAS Workshop on Trust in Agent Societies, May 2016.
12. Günay, A., Liu, Y., & Zhang, J. (2016). Promoca: Probabilistic modeling and analysis of agents in commitment protocols. *Journal of Artificial Intelligence Research*, *57*, 465–508.
13. Günay, A., Songzheng, S., Liu, Y., & Zhang, J. (2015). Automated analysis of commitment protocols using probabilistic model checking. In *Twenty-ninth AAAI conference on artificial intelligence*.
14. Gurobi: Gurobi 8.1. http://www.gurobi.com/products/gurobi-optimizer.

15. Hansen, E. A. (1998). *Finite-memory control of partially observable systems*. Ph.D. Thesis, University of Massachusetts Amherst.
16. Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, *8*(3), 223–250.
17. Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1–2), 99–134.
18. Maheswaran, R. T., Szekely, P., Becker, M., Fitzpatrick, S., Gati, G., Jin, J., et al. (2008). Predictability & criticality metrics for coordination in complex environments. In *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems* (Vol. 2, pp. 647–654).
19. Mallya, A. U., & Huhns, M. N. (2003). Commitments among agents. *IEEE Internet Computing*, *7*(4), 90–93.
20. MATLAB: MATLAB optimization toolbox. https://www.mathworks.com/products/optimization.html.
21. Meneguzzi, F., Magnaguagno, M. C., Singh, M. P., Telang, P. R., & Yorke-Smith, N. (2018). Goco: Planning expressive commitment protocols. *Autonomous Agents and Multi-Agent Systems*, *32*(4), 459–502.
22. Meneguzzi, F., Telang, P. R., & Yorke-Smith, N. (2015). Towards planning uncertain commitment protocols. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems* (pp. 1681–1682).
23. OPTI: OPTI toolbox v2.2. https://www.inverseproblem.co.nz/OPTI.
24. Pereira, R. F., Oren, N., & Meneguzzi, F. (2017). Detecting commitment abandonment by monitoring sub-optimal steps during plan execution. In *Proceedings of the 16th conference on autonomous agents and multiagent systems* (pp. 1685–1687).
25. Poupart, P., Malhotra, A., Pei, P., Kim, K. E., Goh, B., & Bowling, M. (2015). Approximate linear programming for constrained partially observable Markov decision processes. In *Twenty-ninth AAAI conference on artificial intelligence*.
26. Puterman, M. L. (2014). *Markov decision processes: Discrete stochastic dynamic programming*. Hoboken: Wiley.
27. Raffia, H. (1982). *The art and science of negotiation*. Cambridge: Harvard University Press.
28. Sandholm, T., & Lesser, V. R. (2001). Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, *35*, 212–270.
29. Santana, P., Thiébaux, S., & Williams, B. (2016). RAO*: An algorithm for chance-constrained POMDP's. In *Thirtieth AAAI conference on artificial intelligence*.
30. Shiryaev, A. N. (1963). On optimum methods in quickest detection problems. *Theory of Probability & Its Applications*, *8*(1), 22–46.
31. Singh, M. P. (1999). An ontology for commitments in multiagent systems. *Artificial Intelligence in the Law*, *7*(1), 97–113.
32. Singh, M. P. (2012). Commitments in multiagent systems: Some history, some confusions, some controversies, some prospects. In *The goals of cognition. Essays in honor of Cristiano Castelfranchi* (pp. 601–626). London.
33. Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, *21*(5), 1071–1088.
34. Smith, T., & Simmons, R. (2004). Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on uncertainty in artificial intelligence* (pp. 520–527).
35. Sultan, K., Bentahar, J., & El-Menshawy, M. (2014). Model checking probabilistic social commitments for intelligent agent communication. *Applied Soft Computing*, *22*, 397–409.
36. Telang, P. R., Meneguzzi, F., & Singh, M. P. (2013). Hierarchical planning about goals and commitments. In *Proceedings of the 2013 international conference on autonomous agents and multiagent systems* (pp. 877–884).
37. Vokrínek, J., Komenda, A., & Pechoucek, M. (2009). Decommitting in multi-agent execution in non-deterministic environment: Experimental approach. In *8th international joint conference on autonomous agents and multiagent systems* (pp. 977–984).
38. Winikoff, M. (2006). Implementing flexible and robust agent interactions using distributed commitment machines. *Multiagent and Grid Systems*, *2*(4), 365–381.
39. Witwicki, S. J., & Durfee, E. H. (2009). Commitment-based service coordination. *International Journal of Agent-Oriented Software Engineering*, *3*(1), 59–87.
40. Xing, J., & Singh, M. P. (2001). Formalization of commitment-based agent interaction. In *Proceedings of the 2001 ACM symposium on applied computing* (pp. 115–120).
41. Xuan, P., & Lesser, V. R. (2000). Incorporating uncertainty in agent commitments. In *Intelligent agents VI. Agent theories, architectures, and languages* (pp. 57–70). Springer.

42. Zhang, Q., Durfee, E. H., & Singh, S. (2018). Challenges in the trustworthy pursuit of maintenance commitments under uncertainty. In *Proceedings of the 20th international trust workshop co-located with AAMAS/IJCAI/ECAI/ICML 2018* (pp. 75–86).
43. Zhang, Q., Durfee, E. H., Singh, S., Chen, A., & 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* (pp. 3315–3323).
44. Zhang, Q., Singh, S., & Durfee, E. (2017). Minimizing maximum regret in commitment constrained sequential decision making. In *Twenty-seventh international conference on automated planning and scheduling* (pp. 348–356).